■ Server di posta a modo tuo con SquirrelMail ■ I segreti del terminale con Android

LINUX PRO

MAGGIO 2015

Diventa protagonista della programmazione Open Source

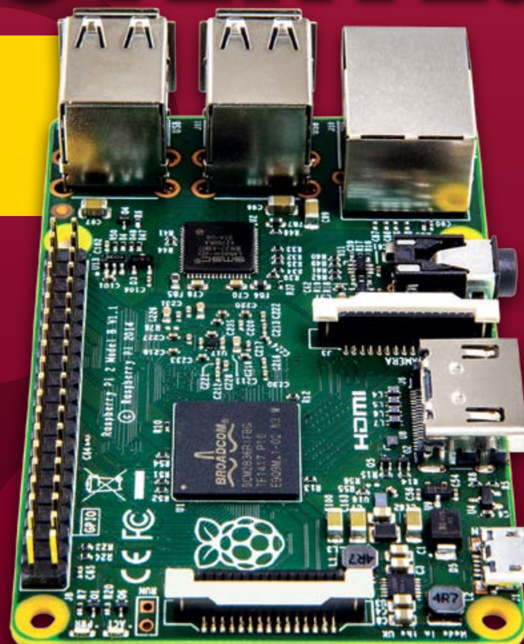


Fedora 21
Ecco le novità

RASPBERRY PI 2
SEMPRE PIÙ POTENTE!



500%
PIÙ VELOCE





Come programmare il server?
Scopri quale linguaggio di scripting usare per configurazioni a prova di bomba **p.44**

Basta intrusioni!
Ecco come difendere il sistema da pericolosi attacchi alla sicurezza **p.58**



TARIFFA R.O.C. - POSTE ITALIANE SPA - ABB. POST. - D.L. 353/2003 (CONV. IN L. 27/02/2004 N. 46) ART. 1, COMMA 1, S/NA - DISTRIBUTORE: PRESS-DI, MILANO

ACCADEMIA DEL CODICE
AC
8 pagine di pura programmazione
da pagina 80



it's a marmo project!

È QUANDO TI SENTI PICCOLO CHE SAI DI ESSERE DIVENTATO GRANDE.

A volte gli uomini riescono a creare qualcosa più grande di loro. Qualcosa che prima non c'era. È questo che noi intendiamo per innovazione ed è in questo che noi crediamo.
Una visione che ci ha fatto investire nel cambiamento tecnologico sempre e solo con l'obiettivo di migliorare il valore di ogni nostra singola produzione.
È questo pensiero che ci ha fatto acquistare per primi in Italia impianti come la rotativa Heidelberg M600 B24. O che oggi, per primi in Europa, ci ha fatto introdurre 2 rotative da 32 pagine Roto-Offset Komori, 64 pagine-versione duplex, così da poter soddisfare ancora più puntualmente ogni necessità di stampa di bassa, media e alta tiratura.
Se crediamo nell'importanza dell'innovazione, infatti, è perché pensiamo che non ci siano piccole cose di poca importanza.
L'etichetta di una lattina di pomodori pelati, quella di un cibo per gatti o quella di un'acqua minerale, un catalogo o un quotidiano, un magazine o un volantone con le offerte della settimana del supermercato, tutto va pensato in grande.
È come conseguenza di questa visione che i nostri prodotti sono arrivati in 10 paesi nel mondo, che il livello di fidelizzazione dei nostri clienti è al 90% o che il nostro fatturato si è triplicato.
Perché la grandezza è qualcosa che si crea guardando verso l'alto. Mai dall'alto in basso.

AGB | artigraficheBocciaspa | A DIFFERENT IMPRINTING.

ARTI GRAFICHE BOCCIA – SALERNO | ROMA | MILANO | PARIS | LONDON | LAUSANNE



CONTACT:
Via Tiberio Claudio Felice, 7 - 84131 Salerno (ITALY)
Tel. +39 089 303311 - Fax +39 089 771017
www.artigraficheboccia.com - info@artigraficheboccia.com

I nemici della sicurezza siamo noi stessi

Un paio di settimane fa abbiamo avuto in redazione una lunga discussione sulla sicurezza, su come proteggere adeguatamente i dati e su come verificare le connessioni. La discussione è in parte nata dall'articolo sul controllo delle porte di un server che pubblichiamo su questo numero. La conversazione è rapidamente degenerata in una corsa ai sistemi più sicuri, da metodi di criptazione dei dati per violare i quali servirebbero eoni a un hacker fino a password così lunghe da non essere memorizzabili, passando per SSL ed eliminazione sicura dei file. A leggere le notizie del settore c'è effettivamente da spaventarsi: fior di grandissime aziende si vedono su base regolare fregare interi database di password (Sony fra tutte) e i servizi di verifica delle password online mostrano quanto sia tutto sommato semplice decriptare dati quando la password usata è troppo breve. Eppure la vera minaccia ai nostri dati e ai nostri server non arriva dai più sofisticati sistemi di aggressione ma... da noi! Una semplice analisi delle varie intrusioni e delle più famose "rapine di dati" mette in evidenza come il vero problema non sia tanto scegliere le giuste procedure ma seguirle. Uno dei furti di dati più grandi, perpetrato ai danni di un colosso del software di creatività, ha messo in mostra come le password degli utenti fossero protette in maniera rudimentale e come i suggerimenti delle password (quelli che usiamo per recuperarle quando le dimentichiamo) fossero salvati in un database in chiaro! Molti utenti usano quei suggerimenti in modi come "La mia parola chiave è pippo", quindi serve a poco criptare la password nel database... Per non parlare di Sony, la cui perdita di dati sarebbe da imputare almeno parzialmente al fatto che le password di accesso dei dipendenti fossero salvate in un file Excel accessibile liberamente in Rete... Insomma, controlliamo le nostre porte e proteggiamo i dati con i sistemi più moderni ma non dimentichiamo mai che l'errore umano è chiave e che ogni struttura di protezione è tanto debole quanto il suo elemento più vulnerabile...

La redazione di Linux Pro





CONTATTI

Domande alla redazione: **redazione@linuxpro.it**
Abbonamenti: **abbonamenti@linuxpro.it**
Arretrati: **arretrati@linuxpro.it**
Problemi con il DVD: **aiutocd@sprea.it**
Sito Web: **www.linuxpro.it**
Oppure inviate le vostre lettere a:
***Linux Pro*, Sprea S.p.A.,**
Via Torino 51, 20063 Cernusco S/N
Telefono: 02.92432.1

LINUX PRO 153

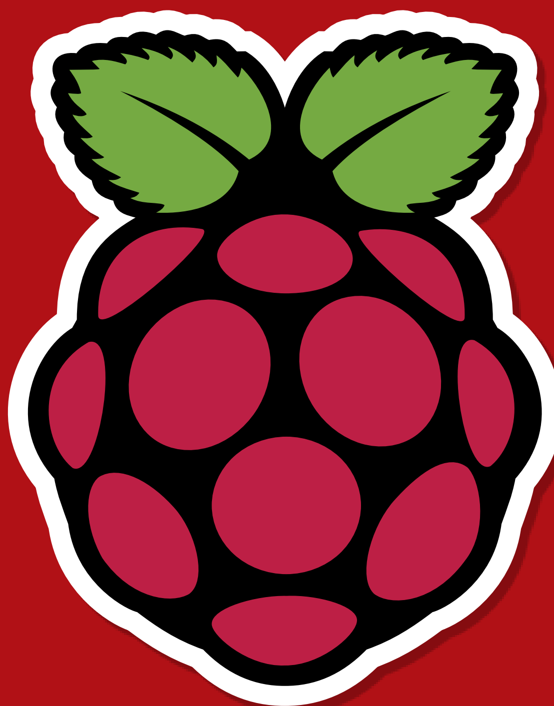
Sommario LINUX PRO

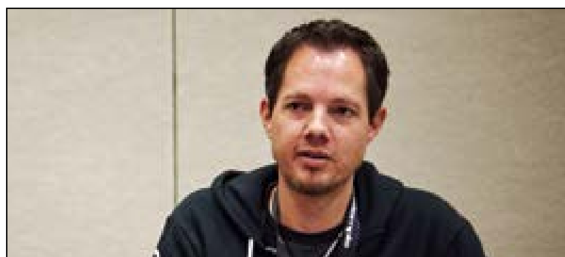
Benvenuti nel centocinquantatreesimo numero di Linux Pro, la guida definitiva a Linux e al mondo Open Source

In primo piano

CHE BELLA LA RPI 2!









Sommario

IL PROSSIMO NUMERO ESCE IL 12 GIUGNO

IL DVD IN BREVE

LATO A

DISTRIBUZIONI
- Fedora 21

LATO B

DISTRIBUZIONI
- Linux Console 2.3
- LinHES 8.3
- SystemRescueCD 4.5

RIVISTA
- Audex 0.79
- Bash 4.3.30
- Chocolate Doom 2.1.0
- ClamAV 0.98.6
- Debreate 0.7.10
- Fifth 0.2
- GeexBox 3.1
- GParted Live 0.22
- Hello 2.10
- Lightweight Sec. 1.5.6
- RecomXaraLX 0.7
- newLISP 10.6.2
- Perl 5.20.2
- Python 3.5.0a3
- QuiteRSS 0.17.6
- Red Eclipse 1.5
- Untangle NG Firewall
- xDecoration 0.1.7
- Xfdesktop 4.10.3

Quando trovi questo bollo negli articoli, cerca il software nel DVD





Newsdesk

Ogni mese tutte le novità dal mondo delle aziende e della comunità Open Source

Se hai news da segnalarci o dei commenti scrivi a newsdesk@linuxpro.it

Mobile

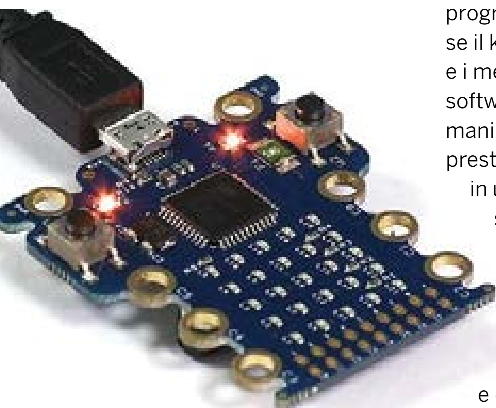
Trent'anni di Software Libero



Stallman festeggia 30 anni di Software Libero: ecco qual è l'importanza della sua missione...

Il 30 marzo di quest'anno è stata una ricorrenza molto importante per i fan dell'Open Source. Infatti, sono ricorsi 30 anni da quando un giovane sviluppatore di software di nome Richard Stallman ha pubblicato il suo Manifesto GNU sul Journal of Software Tools. Stimolato dalla sua naturale diffidenza per il sempre crescente sviluppo del software proprietario, Stallman ha scelto un periodo di forti contrasti per farsi avanti. Infatti, molte macchine Unix venivano vendute senza includere i sorgenti, così come parecchio software proprietario impediva agli sviluppatori di accedere a parti fondamentali del sistema. Stallman aveva una forte preoccupazione che purtroppo, per molti aspetti, si è poi concretizzata nella realtà. L'allora giovane programmatore pensava che se il know-how, la tecnologia e i metodi per sviluppare software fossero caduti nelle mani di poche aziende, ben presto ci saremmo ritrovati in un mondo monopolizzato solo da alcuni nomi, lasciando al di fuori della cerchia tutti coloro che invece aspiravano a un destino di libertà e completa apertura del software. Come detto, queste non erano semplici paranoie, ma previsioni di ciò che Microsoft, Apple e Google avrebbero fatto nel tempo. Stallman decise quindi di varare una precisa strategia per contrastare questo fenomeno. Secondo il suo pensiero, infatti, un utente che vuole apportare cambiamenti in un sistema deve essere libero di farlo. Se non è in grado personalmente, niente gli deve impedire di assumere qualcuno che lo faccia per lui. Gli utenti, in definitiva, non devono essere in balia delle società che possiedono a scatola chiusa i sorgenti. Stallman ha infatti immaginato i software GNU come interamente controllati dalle persone e non viceversa. Da qui è poi nato il suo manifesto, secondo cui solo con un Software Libero si sarebbe potuto continuare a usare la tecnologia senza alcun disonore. Da allora, Stallman è comunque riuscito a concretizzare i suoi obiettivi e non solo per sé stesso, ma anche per milioni di estimatori in tutto il mondo. Infatti, le cifre dei computer che utilizzano Software completamente Libero sono in continua crescita. E per capire quanto gli inglesi comprendano l'importanza della cosa, la BBC (la RAI inglese), per festeggiare i 30 anni della ricorrenza, ha pensato di regalare ben un milione di micro computer agli studenti delle scuole secondarie! Si tratta di sistemi Micro Bit PC molto simili al Raspberry Pi, capaci di colmare le carenze in fatto di informatizzazione delle nuove generazioni. Infatti, questi piccoli dispositivi possono essere utilizzati dai bambini per imparare i rudimenti della programmazione, incoraggiandoli poi a mettere in pratica il codice creato. La speranza è che prima o poi anche l'Italia arrivi a capire, tramite le sua istituzioni, l'importanza del Software Libero e avvii iniziative come quella della BBC. LXP

> **"Devono essere sempre le persone a controllare il software, non viceversa"**



› La BBC ha regalato un milione di Micro Bit ai bambini delle scuole secondarie della Gran Bretagna

Hardware

Ubuntu e l'America Latina

Più macchine Debian vedono la luce nel sud del continente

I cittadini di Colombia, Ecuador e Perù sono ora in grado di entrare in un negozio di informatica e acquistare un PC portatile o un desktop con una distribuzione Linux preinstallata. Dell, infatti, ha lanciato le sue macchine Ubuntu in oltre 500 esercizi commerciali dell'America Latina. Non si tratta solo di una singola operazione commerciale, ma anche di una profonda iniziativa legata alla formazione. Gli addetti, infatti, sono stati istruiti per fornire consulenza e aiuto sia ai clienti sia alle società che intendono passare al mondo Linux. Il personale, inoltre, avrà a disposizione diverse demo che consentiranno di dare uno spaccato completo sulle potenzialità del sistema operativo. Lo sviluppo di questa iniziativa non è isolato. Infatti, deriva dal successo già avuto da Dell in altri mercati, come quello asiatico e indiano. Il parco dei prodotti conterrà diversi computer, tra cui la workstation Dell M3800 Mobile, fornita di specifiche davvero impressionanti: Intel Core i7 da 2,3 GHz, display da 15,6 pollici Ultra HD con supporto per i comandi touch, 16 GB di RAM e SSD da 256 GB. Altri modelli di punta saranno poi l'Inspiron 14 3000 che viene fornito con una configurazione meno muscolosa, ma non per questo scadente: 4 GB di RAM e display da 14 pollici in Full HD. Da parte nostra, non possiamo che essere contenti di questa politica di Dell. Fornire macchine con Linux preinstallato è sicuramente un'iniziativa importante per attrarre ancora più persone verso il mondo Linux. La distribuzione montata è Ubuntu, ma non si esclude che presto se ne possano aggiungere altre. LXP



› **Il Dell M3800 Mobile Workstation, con i suoi 16 GB di RAM, è un vero mostro di potenza**

Sicurezza

Occhio a OpenSSL

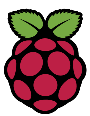
NCC Group verifica la libreria per evitare falle nella sicurezza

Probabilmente **OpenSLL** è la libreria dedicata alla sicurezza più importante del mondo, ma la sua reputazione ha recentemente subìto un duro colpo a causa di alcune falle scoperte nel proprio profilo. Per aiutare a ripristinare la fiducia degli utenti in OpenSLL, così da garantire una maggiore solidità della struttura, il gruppo **NCC** (**https: //us.nccgroup.trust**) si è impegnato in una profonda revisione del codice, con il finanziamento dato dalla Fondazione Linux (CII). NCC si concentrerà sullo stack TLS, così come sugli algoritmi di crittografia e altre parti importanti del codice. **FREAK**, per esempio, (*Factoring attack on RSA-EXPORT Keys*) è un tipo di attacco utilizzato per prendere di mira siti Web di alto profilo, come quelli di American Express, della Casa Bianca o dell'FBI statunitense. Il principio di funzionamento di questa minaccia che ha colpito OpenSLL è costringere i browser che visitano i siti Web a sfruttare un tipo di crittografia piuttosto debole che consentirebbe agli hacker di rubare informazioni personali. Thomas Ritter, ingegnere specializzato in sicurezza del NCC, ha riferito a ZDNet che "NCC è felice di annunciare una verifica completa di OpenSLL. Considerando l'importanza di questa libreria e l'opportunità di immergersi in uno dei software più conosciuti e apprezzati del mondo, il gruppo è onorato di essere stato scelto per questo compito". Speriamo che questa attività porti a dei risultati concreti quanto prima, ristabilendo la giusta fiducia che gli utenti dovrebbero avere in un'infrastruttura di sicurezza come quella garantita da OpenSLL. LXP



I segreti della... Raspberry Pi 2

Spinti dalla curiosità, abbiamo messo subito all'opera la nuova versione di questa scheda



La nuova Raspberry Pi ha decisamente molta più potenza dei suoi predecessori. Questo grazie all'uso di un processore quad-core ARMv7 a 900 MHz e alla memoria molto più generosa del passato, pari a 1 GB. Di questa maggior potenza di calcolo e capacità di memoria trarranno sicuramente beneficio i progetti più esosi di risorse, come OpenCV e Minecraft.

La **Raspberry Pi 2** si porta dietro anche i miglioramenti nel design che avevamo visto nel modello B+, con un maggior numero di porte USB grazie al nuovo chip LAN9514 che fornisce il supporto a quattro porte al posto delle due del LAN9512. La RP B+ aveva introdotto anche una migliore gestione dell'alimentazione e anche la nuova RP ne trae beneficio. "I consumi energetici della RP 2 durante l'esecuzione di un determinato task sono confrontabili con quelli della B+", ci spiega Eben Upton, CEO di Raspberry Pi Trading. "Ovviamente se spingete di più la RP 2 il suo consumo energetico aumenterà, poiché compie un lavoro maggiore. I consumi della B+ a pieno carico sono molto simili a quelli del vecchio modello B, invece". Viste le premesse, non appena ricevuto in redazione un esemplare della nuova scheda, l'abbiamo subito messa al lavoro. Per farlo abbiamo usato la versione più recente di Raspbian presente sul sito della RP (**www.raspberrypi.org/downloads**) visto che essa è dotata del kernel adatto a gestire la nuova CPU ARM Cortex-A7. Il metodo più semplice per scrivere il sistema operativo su una scheda microSD è ricorrere a **NOOBS** (**https://github.com/raspberrypi/noobs**). Serve una scheda microSD da almeno 8 GB: si scarica il file zip di NOOBS dal sito della RP e lo si decomprime nella scheda di memoria, che

> **"La Raspberry Pi 2 sfrutta tutte le migliorie che erano state progettate per la versione B+"**



deve essere formattata in FAT32. Terminata la copia, si smonta la microSD e la si inserisce nel suo slot nella RP2. Si collegano poi il monitor via HDMI, una tastiera e un mouse alle porte USB e un cavo di rete alla porta Ethernet per la connessione a Internet (in alternativa si può usare una chiave USB Wi-Fi). Fatto ciò basta attaccare l'alimentatore alla porta microUSB e la RP2 si avvierà per la prima volta. NOOBS, essendo al primo avvio, chiede quale sistema operativo installare, Raspbian nel nostro caso: lo si seleziona dal menu dopodiché ci vogliono circa 10 minuti per la sua effettiva installazione. Al termine la RP si riavvia e già qui si notano le prime differenze rispetto alle precedenti Raspberry Pi: il boot impiega circa 17 secondi, contro i 33 della RP+. Se poi, durante l'avvio si fa attenzione, si nota la presenza di quattro "lamponi" sullo schermo, a rappresentare i quattro core del nuovo SoC. Terminato il boot Raspbian presenta subito il menu di **raspi-config**: si tratta di un tool per la configurazione della RP che chiameremo in causa dopo. Quindi usciamo da questo tool ed effettuiamo il login come di norma. Le credenziali d'accesso sono le stesse di sempre:

```
USERNAME: pi
PASSWORD: raspberry
```

Inseriti questi dati, si fa partire l'ambiente grafico con

```
startx
```

Il desktop appare leggermente diverso rispetto al passato, grazie ai notevoli cambiamenti fatti dalla Raspberry Pi Foundation nel dicembre dello scorso anno, in gran parte per merito del lavoro di Simon Long che è stato assunto proprio per migliorare l'interfaccia utente. A questo punto la nuova e potente scheda è pronta all'uso, ma come verificarne le prestazioni? Ce la fa, per esempio, a calcolare il valore di Pi greco fino alla decimillesima cifra decimale? Per scoprirlo abbiamo installato una calcolatrice di precisione aprendo un terminale (LXTerminal) e digitando

```
sudo apt-get update
sudo apt-get install bc
```

Poi abbiamo fatto partire il test misurando il tempo impiegato per il calcolo:

```
time echo "scale=10000; 4*a(1)" | bc -l
```

Nel nostro test ci sono voluti 17 minuti e 25,725 secondi; ripetendo l'esperimento con una Raspberry Pi B+ il tempo richiesto per il calcolo è passato a 25 minuti e 5,989 secondi. Questo è un semplice esempio, ma vi fa intuire la maggior potenza del nuovo modello. Come al solito, però, non ci siamo fermati alle prestazioni "out of the box", ma abbiamo cercato di portare al limite la nuova RP usando il tool raspi-config di cui abbiamo parlato prima. Sempre in un terminale, digitate

```
sudo raspi-config
```

› **Il calcolo del valori di Pi greco fino alla decimillesima cifra decimale è un compito impegnativo per la Raspberry Pi 2, ma il tempo richiesto si riduce se provate a overcloccare il processore**

Metti il turbo alla RP

Abbiamo deciso di vedere, per prima cosa, come viene divisa la memoria presente a bordo tra la GPU (*Graphical Processing Unit*) e il sistema principale. Questo dato si scopre scegliendo l'opzione **Memory Split** nel menu delle opzioni avanzate di raspi-config. Sulla Raspberry Pi, di norma, vengono assegnati 64 MB di RAM alla scheda grafica mentre la parte restante viene usata dal sistema. Per l'uso normale, si può lasciare la situazione così com'è; la regola generale è che un terminale non richiede la stessa RAM necessaria a un ambiente grafico, quindi con un progetto che usa solo l'interfaccia a caratteri si potrebbero allocare anche solo 16 MB alla GPU, mentre per applicazioni desktop come **Minecraft** 64 MB sono il minimo. Noi non abbiamo fatto modifiche, ma se si cambia la quantità di memoria della GPU, bisogna riavviare la RP per rendere effettivo il nuovo setup. Il secondo menu »

Tip

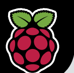


Il browser di default di Raspbian, **Midori**, di recente è stato rimpiazzato con **Epiphany**, che è stato ottimizzato per l'uso con la RP. Il nuovo browser è disponibile tramite l'ultimo aggiornamento di Raspbian e funziona molto bene sia con la RP2 che con le vecchie RP.

Ubuntu sulla RP?

La più grande sorpresa che abbiamo avuto dopo l'uscita del nuovo modello di RP è l'arrivo di una versione di Ubuntu compatibile con essa. Prima del rilascio della RP originale, a inizio 2012, si leggeva spesso in Rete che Ubuntu sarebbe stata disponibile per questo computer. Visto però che Canonical non supporta l'architettura ARMv6, usata dalla CPU ARM11 delle prime RP, sono dovute subentrare altre distribuzioni. Abbiamo visto la nascita di Pidora, un fork di Fedora, usata agli inizi per dimostrare le potenzialità della nuova scheda. Però Pidora cercava di far funzionare un desktop completo su un hardware che richiedeva un ambiente molto leggero. Alla fine fu Debian, sottoforma di Raspbian, a dimostrarsi la soluzione migliore e divenne quindi la distro ufficiale usata nei tutorial online e nella documentazione. L'arrivo della compatibilità con Ubuntu, comunque, non vuol dire che la Raspberry Pi Foundation stia abbandonando Raspbian, tanto che Eben Upton ha affermato che "non stiamo pianificando un'immagine ufficiale di Ubuntu". Eben aggiunge: "Stiamo verificando le prestazioni del port armhf di Debian e di quelle di Raspbian e potremmo fare il cambio se l'aumento di prestazioni ce ne darà motivo. Preferiremmo rimanere su Raspbian, magari con la possibilità di scambio dinamico di alcune librerie chiave, cosa che ci consentirebbe di creare un'unica immagine in grado di supportare sia la RP 2 sia le versioni classiche. Al momento di andare in stampa non esiste un'immagine ufficiale di Ubuntu per la RP ma potete scaricarne una versione comunitaria dall'URL **http://bit.ly/1NUQpG9**. Nella stessa pagina trovate informazioni su Snappy Ubuntu Core, versione pensata per gli sviluppatori.

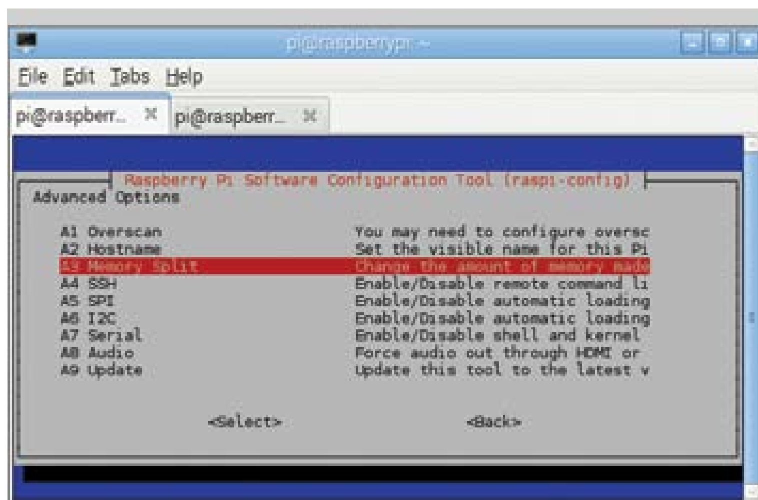
» di raspi-config che abbiamo usato è **Overclock**. I core della RP2 hanno una frequenza di clock pari a 900 MHz, un più che discreto miglioramento rispetto ai 700 MHz dell'originale CPU ARM11 a singolo core. Parlando con i creatori della RP abbiamo scoperto che si può arrivare a una frequenza di 1,1 GHz ma abbiamo poi deciso di non raggiungere questo limite per essere sicuri che il piccolo computer rimanesse stabile. Dobbiamo ricordarvi che spingendo troppo le prestazioni della RP può sempre capitare che la CPU si danneggi a causa dell'eccessivo calore generato dalla maggior velocità di funzionamento. Il team di sviluppo della RP ci ha confermato che il valore di soglia è di 85°: quando vengono raggiunti, la Raspberry Pi si spegne in automatico per evitare danni. Gli amanti dell'overclock spinto, possono comunque fare esperimenti, con l'accortezza però di dotare la propria RP2 di un dissipatore di calore. Ovviamente, è sempre possibile ripristinare i valori di fabbrica avviando nuovamente raspi-config e scegliendo il giusto clock nel menu Overclock. Quindi, dopo l'ennesimo riavvio, abbiamo provato di nuovo il calcolo del valore di Pi. Questa volta il tempo impiegato è stato di 15 minuti e 28,519 secondi, un miglioramento di oltre due minuti. La Raspberry Pi Foundation ha curato con particolare attenzione lo sviluppo della RP2: "Lo sviluppo della Raspberry Pi 2 è durato un paio d'anni", ci hanno detto Eben Upton e Hollingsworth. In questo tempo è incluso anche l'ideazione del chip BCM2836. "Il primo chip è arrivato all'inizio dello scorso maggio; sul blog ci deve essere un video di me, James e Dom nei laboratori di Broadcom all'una di notte, il giorno dopo l'arrivo del chip, con la demo 'video on a teapot' in esecuzione su Linux che girava in una board di sviluppo Ray. La definizione del design della RP2 è iniziata lo scorso agosto (2014) e siamo passati attraverso tre prototipi prima di arrivare alle prestazioni che volevamo", ci ha raccontato Upton.

Tip 

Grazie a **Youtube_dl** ora è possibile vedere i video di YouTube con il browser della RP. Di solito i filmati di YouTube sono in Flash, ma non c'è un pacchetto Flash per la RP. Quando guardate un video con il browser, Youtube_dl sostituisce gli elementi in Flash con gli analoghi in HTML5.



› **Le demo Hello_Pi sono utilissime per mostrare le potenzialità della RP 2. Potete perfino proiettare un filmato sulla superficie di questa teiera o di un cubo**



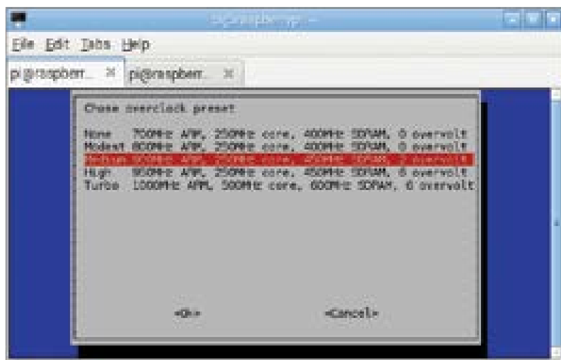
› **Nel menu delle impostazioni avanzate di raspi-config trovate tutte le opzioni per personalizzare il funzionamento del piccolo computer**

Compatibilità

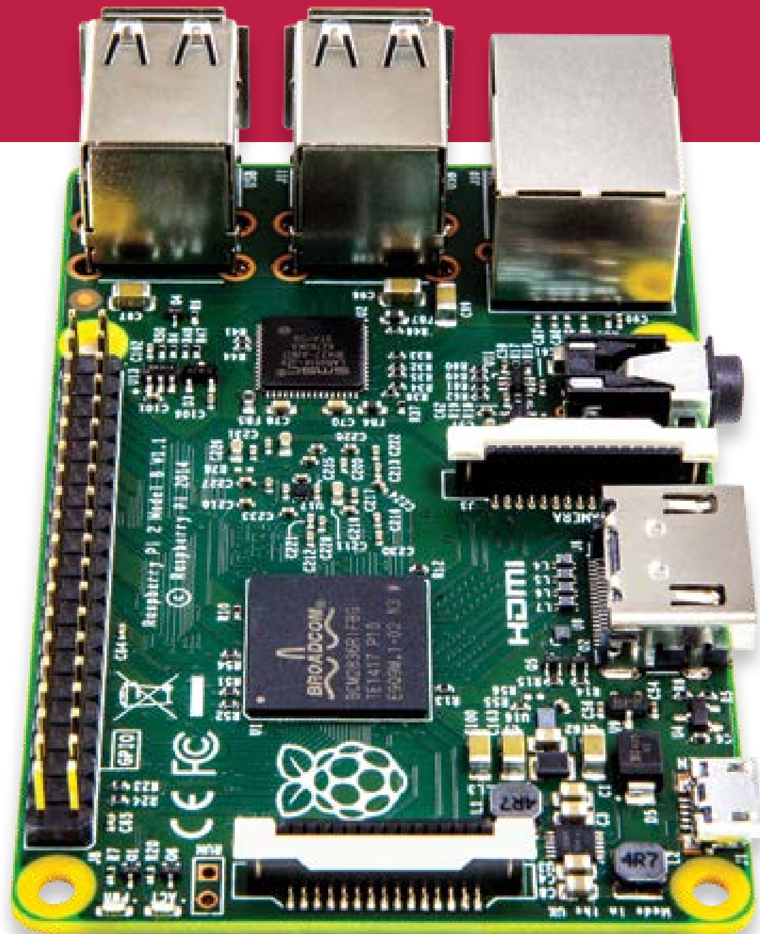
La scelta del nuovo processore, pur portando un aumento delle prestazioni, ha fatto sì che fosse mantenuta comunque la compatibilità con i modelli precedenti. Invece di scegliere un'architettura differente, la RP Foundation ha continuato a usare una CPU ARM compatibile con l'ARM11 usata finora. L'ARMv7 quad core può eseguire senza problemi il software scritto per le vecchie RP: "Raspbian funziona senza cambiare nulla, c'è solo bisogno di un nuovo kernel che è incluso nel file che si scarica dal nostro sito", ci ha detto Eben. Per quel che riguarda la compatibilità hardware, la RP2 ha lo stesso GPIO della RP B+, il che vuol dire che le schede d'espansione realizzate per la A+ e la B+ funzionano anche con la RP 2; questo include anche le schede HAT (*Hardware*

Com'è stata costruita la RP2

I rappresentanti della Raspberry Pi Foundation sono entusiasti del nuovo modello. Abbiamo parlato del suo sviluppo con il team di ingegneri e con Gordon Hollingsworth. "La Raspberry Pi 2 è maestosa. È vicina alle prestazioni di un tipico PC, proprio il punto a cui volevamo arrivare alla nascita del progetto". Lo sforzo fatto per portare avanti il compito è incredibile: "Il team ha messo assieme l'equivalente di 20 anni-uomo di lavoro per la nuova Raspberry Pi e il suo processore, e si può considerare un costo di 2 o 3 milioni di sterline". Nonostante questo entusiasmo, i vecchi modelli non sono stati abbandonati. Eben ci ha spiegato che: "Ci sono molti clienti industriali che non passeranno al nuovo modello e, ovviamente, abbiamo ancora la versione A+. Per darti un'idea della pervasività della RP classica, abbiamo venduto circa 80.000 esemplari della B dopo il lancio della B+". La fondazione ha prodotto anche il Compute Module, che è stato creato per includere la RP nelle applicazioni industriali. Abbiamo chiesto se anche questo modulo verrà aggiornato: "Prima o poi faremo un Compute Module 2, ma non nella prima metà del 2015". E per quel che riguarda la A+, verrà aggiornata? "Non abbiamo pianificato nulla per questo modello, poiché il suo prezzo è così limitato da non consentire una modifica importante mantenendo fermo il costo". In tutti i casi la famiglia della Raspberry Pi è cresciuta parecchio dal 2012 e ora conta ben sei dispositivi.



› **Il menu *Overclock* di raspi-config vi permette di spingere le prestazioni della nuova RP, ma occhio a non esagerare**



Attached on Top) che contengono un chip che comunica con la RP per eseguire automaticamente il setup. In realtà, però, ci sono alcune schede che, a causa delle dimensioni o del design, non sono interfacciabili con la B+ o la RP 2; per esempio prodotti come PiFace e PiFace Control and Display. Comunque il team di OpenLX SP, che sviluppa queste schede, sta rilasciando delle versioni compatibili con la B+ e la RP 2.

Come va con il 3D?

Ogni RP include una GPU chiamata VideoCore IV che si occupa della grafica e arriva a visualizzare filmati in alta definizione a 1080p. Anche la nuova RP ha questa GPU, realizzata sempre da Broadcom proprio come il BCM2836. Abbiamo deciso di verificare anche le prestazioni di questo componente, usando una suite di benchmark per la grafica presente in Raspbian. Se si apre LXTerminal e si digita

```
cd /opt/vc/src/hello_pi/
```

si trovano diverse directory contenenti svariate demo. Prima di poterle usare, però, bisogna compilarle tramite uno script apposito:

```
./rebuild.sh
```

Questo script impiega alcuni minuti a compilare tutta la suite di demo, anche sulla nuova RP, per cui bisogna pazientare un po'. Completata l'operazione abbiamo subito provato alcuni esempi, il primo è stato **hello_teapot**. Per eseguirlo è stato sufficiente spostarsi nella directory omonima e lanciarlo così:

```
cd hello_teapot
./hello_teapot.bin
```

Compare sullo schermo il disegno 3D di una teiera con il video di Big Buck Bunny che viene renderizzato sulla sua superficie. Per uscire da questa demo basta premere **Ctrl**+**C**. Un'altra demo interessante è **hello_triangle2**: per raggiungerla bisogna tornare indietro di un livello nell'albero delle directory, con

```
cd ..
```

Da qui basta entrare nella cartella di hello_triangle2 ed eseguire la demo come fatto con l'altra:

```
cd hello_triangle2
./hello_triangle2
```

All'inizio questa demo sembra statica, ma non appena si prova a muovere il mouse si vedono due frattali sovrapposti che si muovono e reagiscono agli spostamenti del mouse. Anche in questo caso, per terminare la demo, si preme **Ctrl**+**C**, ritornando a LXTerminal. Bene, a questo punto la nostra prima prova della RP 2 è quasi terminata. Nella prossima pagina di questo articolo potete vedere com'è andato il nostro esperimento d'interfacciamento di Minecraft con la scheda Pibrella allo scopo di creare un pulsante per la distruzione globale! 

Tip

La Raspberry Pi 2 ha le stesse dimensioni del modello B+ ma se pensate di riutilizzare alcuni dei case di quest'ultima RP, come il Pibow, dovete considerare che alcuni dei componenti presenti sulla scheda sono stati spostati. Questi cambiamenti non influenzano le dimensioni complessive ma la differente geometria potrebbe impedire il corretto inserimento, in alcuni casi.

Aggiornamenti

La Raspberry Pi Foundation ha rilasciato diversi aggiornamenti della distro Raspbian. Tenere aggiornato il proprio sistema è una buona abitudine e, fortunatamente, è anche facile farlo. Servono solo alcuni semplici comandi da terminale. Si inizia con

```
sudo apt-get update
```

che aggiorna l'elenco dei programmi installabili. Digitando poi

```
sudo apt-get upgrade
```

si installano i pacchetti di cui esiste una nuova versione. Oppure si può usare

```
sudo apt-get dist-upgrade
```

che, oltre ad aggiornare i pacchetti, rimuove anche i vecchi kernel. Se la versione di Raspbian che state usando non ha il nuovo desktop creato da Simon Long, lo potete installare inserendo i tre comandi seguenti nel terminale:

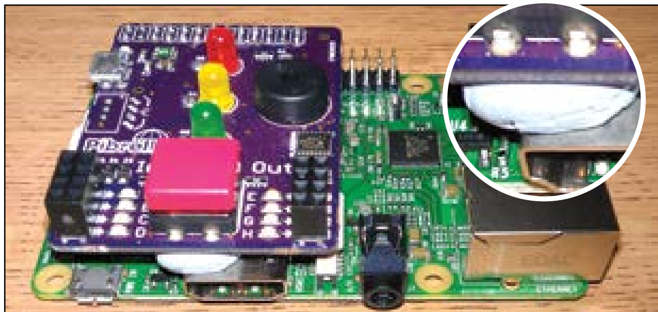
```
sudo apt-get update
sudo apt-get dist-upgrade
sudo apt-get install raspberrypi-ui-mods
```

Riavviate la RP e, al successivo login, vi ritroverete davanti la nuova interfaccia grafica di Raspbian. Ricordiamo che tutti gli aggiornamenti vengono eseguiti grazie al fenomenale gestore di pacchetti **apt**, usato in tutte le distro che derivano da Debian.

Fate comunicare Pibrella e Minecraft

1 Collegate Pibrella alla vostra RP

Vi mostriamo come usare il grosso pulsante rosso di Pibrella per far detonare la TNT in Minecraft. Pibrella si collega ai primi 26 pin del GPIO della RP. Non collegatela mai quando la RP è alimentata. Usate della creta o qualcosa di simile per evitare che si crei un corto circuito con la porta HDMI. Ora collegate i cavi come al solito e inserite l'alimentazione nella porta micro USB.

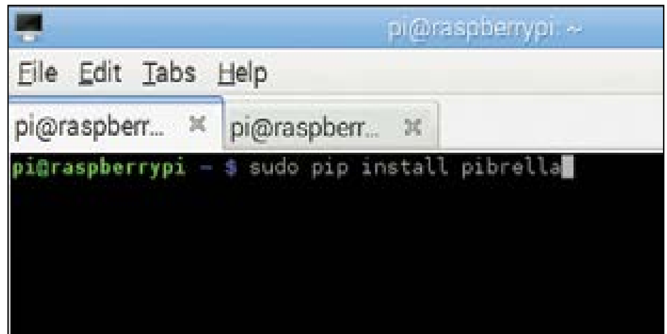


2 Impostate Pibrella

Al termine del boot di Raspbian, aprite LXTerminal e digitate

```
sudo apt-get update
sudo apt-get install python-pip
sudo pip install pibrella
```

Installerete così il necessario per usare Pibrella con Python.

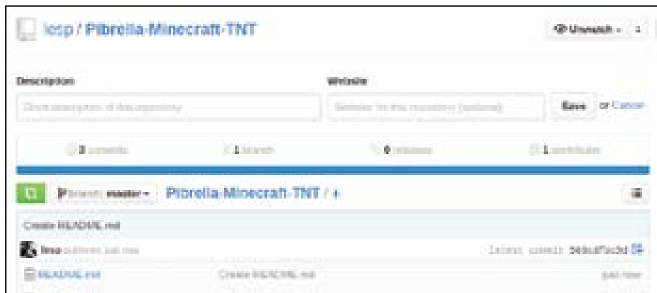


3 Recuperare il codice sorgente

Abbiamo creato un repository su GitHub che contiene il codice per questo tutorial: visitate l'URL **https://github.com/lesp/Pibrella-Minecraft-TNT** e scaricatene una copia. Poi aprite il terminale e scrivete:

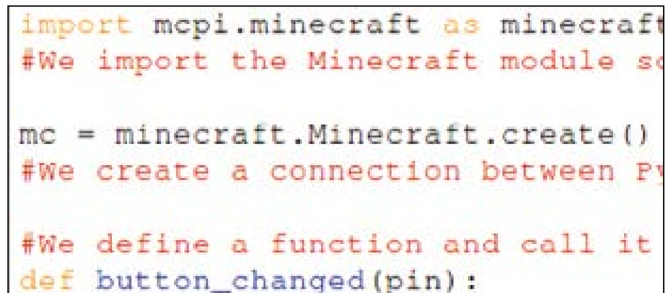
```
sudo idle
```

Si aprirà Idle, un editor per Python, con i privilegi di root, col quale potrete comandare Pibrella. Aprite il codice d'esempio.



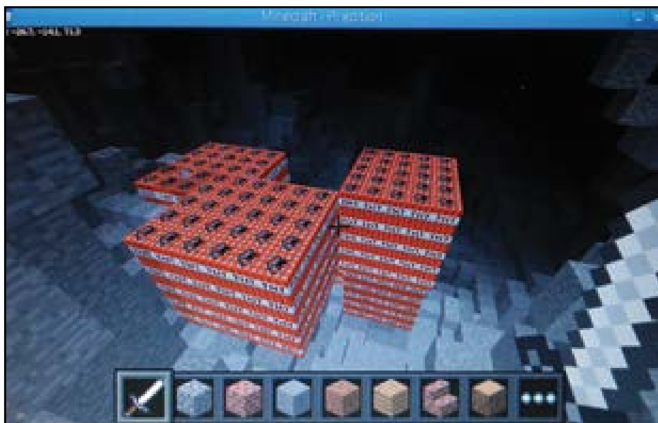
4 Esaminate il programma

Il nostro codice è scritto in Python 2, visto che il modulo per Minecraft è disponibile solo per questa versione; il listato è di facile lettura. Le linee che iniziano con **#** sono i commenti. Le prime righe contengono i comandi per importare le funzionalità extra necessarie, cioè le librerie di Pibrella e Minecraft. Subito dopo abbiamo usato la variabile **mc** per memorizzare i dettagli della connessione con Minecraft.



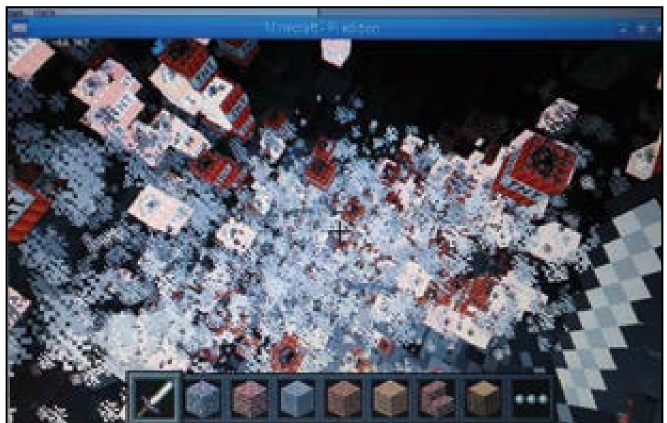
5 In posizione

Minecraft usa le coordinate x,y,z per conoscere la posizione degli oggetti all'interno del suo mondo. Abbiamo creato la funzione **button_changed()** che localizza il giocatore e crea un cubo di TNT nelle sue vicinanze. Infine abbiamo impostato l'esecuzione di questa funzione quando viene premuto il pulsante. Tenete aperta la finestra dell'editor, avviate Minecraft e create un nuovo mondo virtuale.



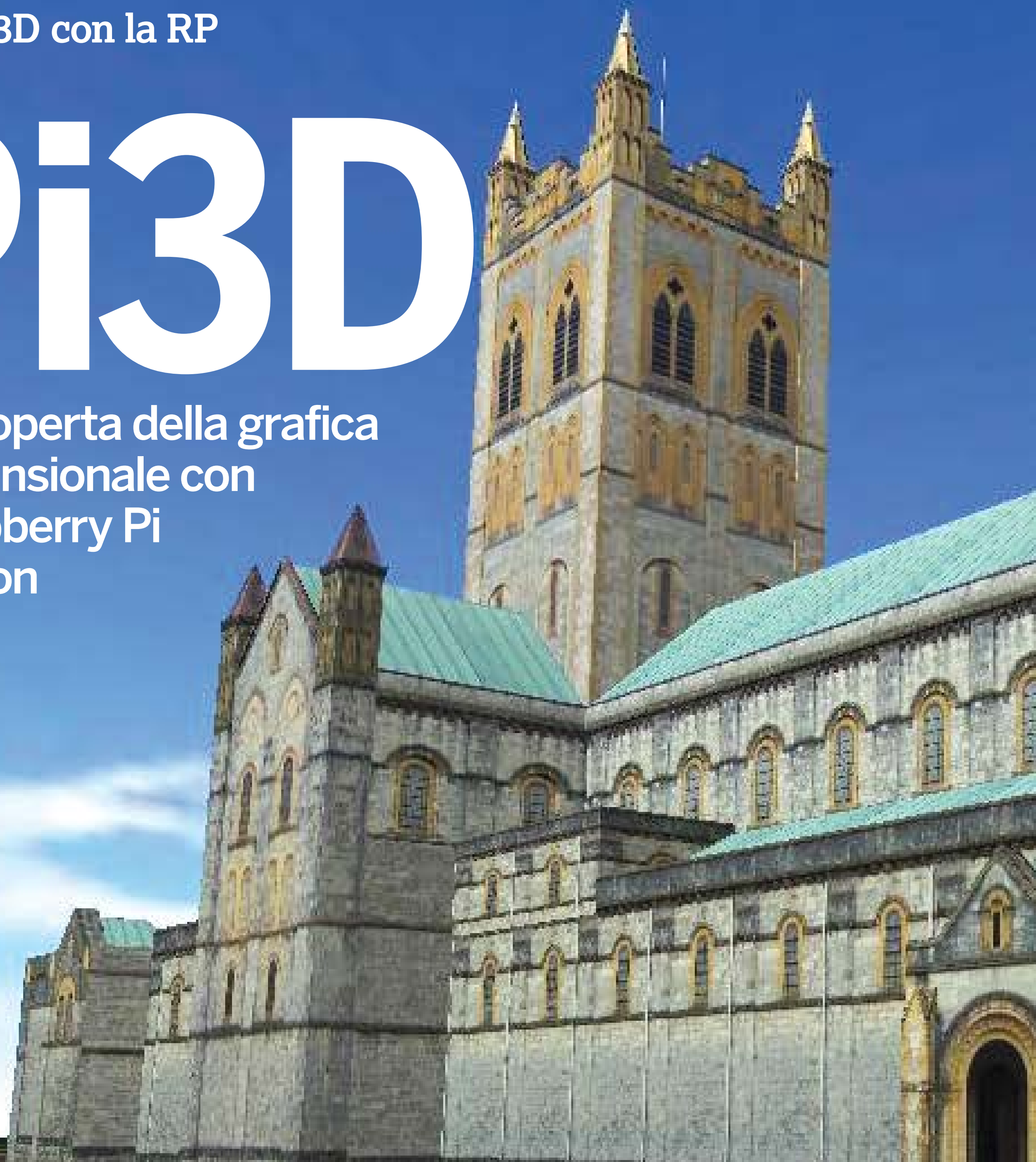
6 Rilasciate la bomba

Con Minecraft pronto e il codice aperto, premete **TAB** per rilasciare il mouse dalla finestra del gioco e, nell'editor Python, andate in **Run** -> **Run Module**: la shell di Idle eseguirà il codice. Tornate in Minecraft e spostatevi in un posto carino. Ora premete il pulsante rosso di Pibrella per rilasciare la bomba. Colpite la TNT con la vostra spada e... scappate! Se usate la vecchia RP, Minecraft potrebbe andare in crash. LXP



Pi3D

Alla scoperta della grafica tridimensionale con la Raspberry Pi e Python



Nonostante le sue ridotte dimensioni e la scarsa potenza di calcolo, la RP gode di un interessantissimo hardware grafico. Si tratta del processore dual core Broadcom VideoCore IV che supporta OpenGL ES 2.0 e la decodifica in hardware dei video H.264 a 1080p e 30fps. Bene... ma cos'è esattamente OpenGL ES e come si può usare per creare grafica 3D? Se vi state chiedendo questo, continuate a leggere e lo scoprirete.

OpenGL ES è un sottoinsieme delle librerie grafiche OpenGL – per fare grafica 2D e 3D – progettato per funzionare nei dispositivi embedded. Esistono diversi binding per molti linguaggi, ma usarli in modo diretto è una cosa riservata ai più bravi (o coraggiosi): programmare in OpenGL non è facile come farlo in Python, tanto per capirci. Fortunatamente, chi usa Python ha a disposizione diverse opzioni per sfruttare tutte le meraviglie della grafica 3D senza doversi adattare a concetti tanto complessi da sembrare alieni. Per esempio, in passato vi abbiamo parlato di **Clutter** (**https://wiki.gnome.org/Projects/Clutter**), una pratica soluzione per usare la grafica mediante concetti familiari come quelli di stage e attori. Clutter può funzionare anche senza un server X, ottima cosa per realizzare delle demo leggere. Questa libreria deve però essere patchata e ricompilata per usare OpenGL ES, come potete leggere sul sito Web della RP: **http://bit.ly/ClutterandCoglonPi**. Vi farà quindi piacere sapere che esiste una via meno impervia per smanettare con la grafica 3D sulla RP grazie al lavoro di tre persone: Tim Skillman, Paddy Gaunt e Tom Ritchford. Nel 2011, ancor prima che il suo progetto fiorisse, Skillman pubblicò brani di codice sperimentale nel forum della RP, ottenendo un gran numero di incoraggiamenti a continuare. Potete leggere l'annuncio ufficiale all'URL **www.raspberrypi.org/pi3d**, ma bando alle ciance, è ora di cominciare. Per prima cosa seguite quanto descritto nel box **Installare Pi3D in Raspbian** e sarete pronti a partire. Il modo migliore per imparare l'uso di Pi3D è dare un'occhiata agli esempi: il nostro preferito è il file **Earth.py** presente nella directory delle demo.



› Il 3D di Pi3D è così realistico che vi verrà naturale cercare di entrare dentro questa abbazia...

Questo esempio vi insegnerà tutto quello che c'è da sapere su sfere, texture, input dalla tastiera e un po' di geometria stellare. Visto che il mondo sta per passare all'uso di Python 3, si inizia con il modulo __future__, importando alcune funzioni che garantiscono la compatibilità. Poi è il turno di alcune funzioni trigonometriche, visto che non si può fare nulla senza seno e coseno, infine c'è il modulo pi3d stesso. Il divertimento inizia con la definizione di un oggetto importante, DISPLAY. Visto che molti altri oggetti si basano sulla sua esistenza, la maggior parte dei progetti realizzati con Pi3D contengono la sua definizione nelle prime righe di codice. Questo oggetto contiene le informazioni di temporizzazione utili per le animazioni. Potete passare la dimensione dello schermo alla funzione **create()**, noi di seguito usiamo una finestra piccola, 50x50, impostando uno sfondo nero opaco:

```
DISPLAY = pi3d.Display.create(x=50, y=50)
DISPLAY.set_background(0,0,0,1) #r,g,b,alpha
```

Questione di shader

Una parte della magia di OpenGL (e di OpenGL ES) si chiama **shader**. Gli shader sono programmi scritti in un... linguaggio shader, nel nostro caso GLSL, e gestiscono i riflessi, le ombre e tutte le altre interazioni tra la luce e le superfici, i volumi e i punti. GLSL ha una sintassi simile a quella del C ed è progettato per sfruttare al massimo il gran numero di unità di shader presenti nelle schede grafiche moderne. L'idea generale è di avere tanti piccoli programmi shader che girano in parallelo e che, tutti assieme, arrivano a creare delle scene complicate ma piacevoli da vedere. Intelligentemente Pi3D nasconde i dettagli più complessi dell'implementazione degli shader, ma questo non vi impedisce di ottenere degli effetti accattivanti per la Terra, la Luna e le stelle:

```
shader = pi3d.Shader("uv_light")
shinesh = pi3d.Shader("uv_reflect")
flatsh = pi3d.Shader("uv_flat")
```

Lo shader **uv_light** usa la direzione della luce e le ombre per creare un effetto 3D, mentre **uv_flat** fa il rendering della texture senza eseguire alcuna trasformazione dei colori. Infine, **uv_reflect** fa rispecchiare un'immagine su un'altra. Nell'esempio viene usata per riflettere la texture della Luna grande su quella più piccola, e anche per rispecchiare le stelle. Questa demo è accompagnata da diverse immagini presenti nella directory **textures**. I file PNG qui presenti contengono informazioni sulla trasparenza, cosa utile visto che tra poco vorremo riuscire a vedere cosa c'è dietro di essi. D'altronde questo è un tutorial sulla grafica 3D. Per esempio, immaginate di voler sovrapporre il file **earth_clouds.png** alla Terra, in modo da dare l'impressione della presenza dell'atmosfera. Nelle righe di codice seguenti, l'argomento **True** nella prima linea indica che bisogna rispettare le informazioni sulla trasparenza parziale presenti nella texture delle nuvole. Prima si caricano tutte le texture:

```
cloudimg = pi3d.Texture("textures/earth_clouds.png",True)
earthimg = pi3d.Texture("textures/world_map.jpg")
moonimg = pi3d.Texture("textures/moon.jpg")
starsimg = pi3d.Texture("textures/stars2.jpg"
watimg = pi3d.Texture("textures/water.jpg")
moonbmp = pi3d.Texture("textures/moon_nm.jpg")
```

Le texture non sono molto utili se non c'è una superficie a cui applicarle. Quindi andrete a disegnare la Terra e la Luna, rappresentate entrambe da sfere. Vi serviranno due sfere per la Terra: una mostrerà i dettagli della superficie, l'altra (con un raggio leggermente superiore) servirà per il rendering delle nuvole e di altri effetti shader. L'esempio include anche due lune, per mostrare un sistema gerarchico di rotazioni nel quale una luna più piccola orbita attorno a una luna più grande, che a sua volta gira attorno alla Terra. Inoltre bisogna specificare anche un piano su cui effettuare il rendering delle stelle. Oltre a indicare il raggio e il centro, il costrutto **Sphere** richiede altri due parametri extra, **slices** e **sides**, che definiscono il numero di fette in direzione latitudinale e il numero di segmenti con cui viene approssimata la sfera. È vero, tuttora non ci siamo evoluti molto oltre i pixel e le linee dritte, quindi non siamo ancora in grado di disegnare una sfera senza approssimazione. Invece c'è un modo facile per gestire l'input da tastiera, che »

Installare Pi3D in Raspbian

Pi3D ha alcune dipendenze, inclusi degli header che troverete nei repository di Raspbian. Quindi, per prima cosa aggiornate il sistema e poi installate ciò che serve:

```
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install python-dev python-setuptools libjpeg-dev zlib1g-dev libpng12-devlibfreetype6-dev
```

Le librerie Pi3D e Pillow non sono presenti nei repository di Raspbian, almeno per ora, per cui dovete ricorrere a un altro strumento per installarle: **Pip**. Procedete con la sua installazione e poi con quella delle librerie che vi servono:

```
$ sudo apt-get install python-pip
$ sudo pip install Pillow
$ sudo pip install pi3d
```

Di default, i modelli con 512 MB di memoria della RP allocano 64 MB alla GPU. Anche se molte delle demo di Pi3D funzionano con questo quantitativo di memoria, alcune ne richiedono di più. Potete modificare la memoria allocata alla GPU usando l'utility **raspi-config**.

```
$ sudo raspi-config
```

128 MB di RAM dovrebbero essere sufficienti e vi consentono di eseguire anche alcune applicazioni desktop. Le demo di Pi3D si possono recuperare da GitHub:

```
$ cd ~
$ git clone git://github.com/pi3d/pi3d_demos
```

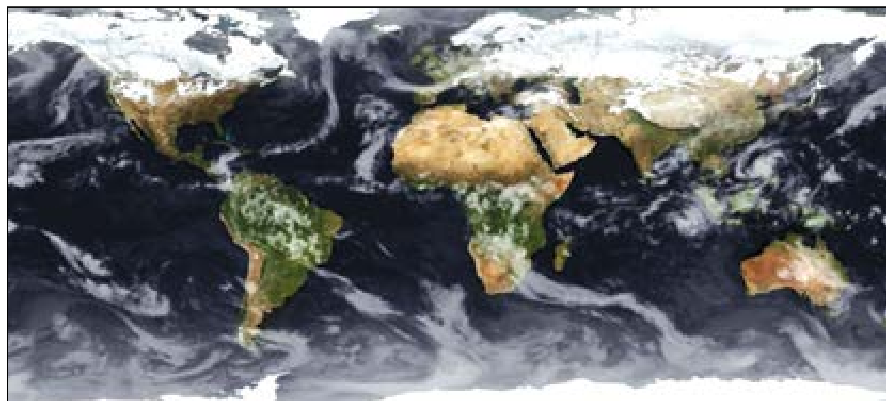
Questo comando creerà la directory **~/pi3d_demos** nella quale potete trovare varie demo, per esempio Buckfast Abbey:

```
$ cd ~/pi3d_demos
$ python BuckfastAbbey.py
```

» in questo caso si usa solo per far terminare il programma:

```
mysphere = pi3d.Sphere(radius=2, slices=24, sides=24, name="earth", z=5.8)
mysphere2 = pi3d.Sphere(radius=2.05, slices=24, sides=24, name="clouds", z=5.8)
mymoon = pi3d.Sphere(radius=0.4, slices=16, sides=16, name="moon")
mymoon2 = pi3d.Sphere(radius=0.15, slices=16, sides=16, name="moon2")
myplane = pi3d.Plane(w=50, h=50, name="stars", z=30)
mykeys = pi3d.Keyboard()
```

Se non specificate le coordinate x, y o z della vostra sfera, esse cadranno nel punto d'origine.



› **Per far piovere vi servono le nubi e per far passare la luce vi servono i canali alpha**

Balletto celestiale

Ora userete i parametri **rot1** e **rot2** per definire, rispettivamente, l'orbita della luna attorno alla Terra e della seconda luna attorno alla prima. Questi parametri verranno incrementati man mano che il loop principale (che vedrete a breve) procede. Il raggio di ogni orbita è definito da **m1Rad** e **m2Rad**. Inizializzando rot1 e rot2, rispettivamente, al valore 90 e 0 si posiziona (secondo il punto di vista di chi guarda) la luna più grande di fronte alla Terra e quella più piccola a fianco di quella più grande.

```
rot1=90.0
rot2=0.0
m1Rad = 4
m2Rad = 0.6
```

Oltre alla rotazione dei satelliti, bisogna anche far ruotare su se stesse – quindi attorno l'asse y – sia la Terra che le due lune. L'asse y corrisponde alla direzione verticale sullo schermo. Qui bisogna applicare un piccolo trucco, facendo ruotare la sfera che "contiene" l'atmosfera un po' più velocemente di quella che rappresenta la Terra, così da ottenere un effetto più carino e veritiero, visto che le nuvole non sono ancorate a terra. Visto poi che tutto ruota, conviene far girare anche lo sfondo stellato: essendo un piano, quindi con due dimensioni, ha senso farlo girare attorno all'asse z. Perciò bisogna ridisegnare le sfere **moon1** e **moon2** cambiando le loro proprietà relative alla posizione, usando della trigonometria d'alta scuola per ottenere le nuove coordinate. L'oggetto **DISPLAY** vi consente di creare un ciclo (loop) che userete per governare questo balletto celestiale:

```
while DISPLAY.loop_running():
  myplane.rotateIncZ(0.01)
  mysphere.rotateIncY(-0.1)
  mysphere2.rotateIncY(-0.14)
  mymoon.position(mysphere.x() + m1Rad*sin(rot1), mysphere.y(), mysphere.z() - m1Rad*cos(rot1))
  mymoon.rotateIncY(-0.1)
  mymoon2.position(mymoon.x() - m2Rad*sin(rot2), mymoon.y(), mymoon.z() + m2Rad*cos(rot2))
  mymoon2.rotateIncZ(-0.61)
```

A questo punto dovete ricorrere agli shader per aggiungere le texture e qualche altro effetto ai corpi celestiali appena creati e animati. Lo shader reflect usato sulle lune, oltre al nome della texture, richiede un paio di parametri numerici che specificano rispettivamente il numero di piastrelle (tile) da usare e la forza della riflessione. Le nuvole devono essere disegnate per ultime, altrimenti l'insieme delle trasparenze non funzionerà: non potete aggiungere successivamente oggetti più lontani oscurati da un altro oggetto semitrasparente quando si specifica **blend = True**, quindi conviene aggiungere la texture delle nuvole per ultima.

```
  mysphere.draw(shader, [earthimg])
  mymoon.draw(shinesh, [moonimg, moonbmp], 6.0, 0.0)
  mymoon2.draw(shinesh, [watimg, moonbmp, starsimg], 3.0, 0.8)
  myplane.draw(flatsh,[starsimg])
  mysphere2.draw(shader, [cloudimg])
```

Ora bisogna incrementare i parametri legati alla rotazione – la luna più piccola gira attorno a quella grande circa quattro volte più velocemente di quanto non faccia la luna grande attorno alla Terra:

```
  rot1 += 0.005
  rot2 += 0.021
```

Visto che avete impostato in precedenza un oggetto Keyboard, sarebbe un peccato non usarlo. Dovrete individuare due eventi da tastiera: premendo **Esc** (identificato dal codice ASCII 27) si deve uscire dal programma, mentre premendo **p** (codice ASCII 112) si deve fare uno screenshot.

Cosa è OpenGL

Di sicuro avrete già sentito parlare più volte di OpenGL (*Open Graphics Library*), un'API (*Application Programming Interface*) per comunicare con l'hardware grafico. Le svariate chiamate a funzione che fanno parte di questa API sono nate nel 1992 in Silicon Graphics, che decise di mettere in Open Source le sue librerie diventate uno standard allo scopo di ottenere un vantaggio sulle aziende concorrenti. La mossa funzionò e la libreria si diffuse ulteriormente, non impedendo però lo sviluppo delle librerie Direct3D di Microsoft. OpenGL, comunque, ha sempre trovato il suo spazio, di questi tempi in gran parte sottoforma di OpenGL ES, il sottoinsieme di OpenGL usato nei dispositivi mobile, nei sistemi embedded e in alcune console da gioco. Diversamente dalle normali schede grafiche dei PC, questi sistemi spesso mancano di alcuni registri e del supporto integrato ai numeri in virgola mobile, quindi le cose devono essere gestite di conseguenza. Il principio, comunque, rimane lo stesso: avere un metodo uniforme per far realizzare all'hardware grafico texture, luci e calcoli per la prospettiva. Così come OpenGL, anche OpenGL ES viene gestita da Khronos Group, un consorzio composto da importanti industrie e istituzioni accademiche. Oltre a OpenGL, questo consorzio coordina lo sviluppo di OpenCL, WebGL, EGL e alcuni altri standard legati alla grafica e al video. Nell'agosto dello scorso anno si è aggiunto un nuovo membro, Microsoft, perché anch'essa riconosce l'importanza di avere uno standard aperto per il mercato dei dispositivi mobile, ed è interesse di tutti che queste regole siano fissate partendo dall'input e dal supporto di tutti gli attori più importanti.

```
k = mykeys.read()
 if k >-1:
   if k==112:
     pi3d.screenshot("earth1.jpg")
   elif k==27:
     mykeys.close()
     DISPLAY.stop()
     break
```

Sfocature

Con quanto detto finora si conclude il codice presente nell'esempio Earth.py; sentitevi liberi di modificare le cose come più vi pare opportuno, per capire come funzionano le varie istruzioni. Fatto questo, provate a seguirci ancora, vi mostriamo di seguito come giocare con la sfocatura e la profondità di campo della Luna, in modo che appaia non a fuoco quando si allontana dal punto d'osservazione (cioè voi che la guardate). Per compiere questa nuova magia dovete evocare il modulo **Defocus**. Piazzate la linea di codice seguente da qualche parte prima del loop principale (per esempio dopo la definizione degli shader):

```
defocus = pi3d.Defocus()
```

L'operazione di sfocatura funziona racchiudendo le normali chiamate a **draw()** all'interno di un blocco delimitato da **start_blur()** ed **end_blur()**. Gli oggetti disegnati all'interno del blocco verranno renderizzati in un buffer e non appariranno sullo schermo. Per renderli visibili si usa il metodo **blur()** che applicherà un effetto sfocatura adeguato al caso, in base alla distanza. Quindi racchiudete le linee che iniziano con **mymoon.draw** in questo modo:

```
  defocus.start_blur()
  mymoon.draw(shinesh, [moonimg,
moonbmp], 6.0, 0.0)
  defocus.end_blur()
```

Il metodo blur, che disegna effettivamente gli oggetti, richiede tre parametri aggiuntivi (oltre al nome dell'oggetto Shape da disegnare): la distanza focale, la distanza dopo la quale tutto sarà sfocato al massimo e il grado di massima sfocatura. Provate a impostare il primo parametro a 0. Poi, visto che l'orbita della Luna ha un raggio pari a 4, assegnate il valore 3 al secondo parametro e, infine, per il terzo numero, provate a fare degli esperimenti. Impostandolo con un valore troppo grande si finisce per avere il fenomeno del banding (cioè i colori vengono rappresentanti in modo errato), vi suggeriamo di provare il valore 5. Il tutto si riassume nella linea seguente, da inserire subito dopo **defocus.end_blur()**:

```
defocus.blur(mymoon, 0, 3, 5)
```



› **Potete importare con facilità i modelli 3D salvati nel formato Panda3D (.egg)**



› **Non avete la vista appannata... la Luna è sfocata... ma che ci fa un altro satellite lì in mezzo?**

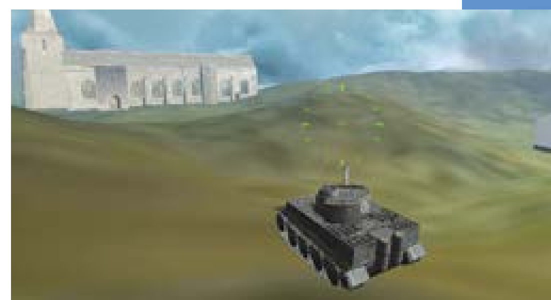
Aggiungi una videocamera

Se volete, il divertimento non finisce qui. Provate ad aggiungere un oggetto Camera e potrete immergervi completamente nel sistema 3D. Usando solo un pizzico di trigonometria e l'oggetto Keys già implementato prima, potete muovere il punto d'osservazione. Dovrete aggiungere la funzione **radians** alle funzioni trigonometriche già importate dal modulo **math**. Poi impostate l'oggetto Camera e inizializzate alcune sue proprietà subito dopo la dichiarazione di DISPLAY:

```
CAMERA = pi3d.Camera()
 rot = 0
 tilt = 0
 rottilt = True
 camRad =5
```

Userete la variabile booleana **rottilt** per attivare qualunque cambiamento alla posizione della camera o al suo orientamento. Ruotare o alzare la camera è facile, ma cambiare il raggio (determinato da **camRad**) richiede un po' di operazioni trigonometriche. Quindi, il loop principale diventa:

```
while DISPLAY.loop_running():
  if rottilt:
    CAMERA.reset()
    CAMERA.rotate(tilt, -rot, 0)
    CAMERA.position(camRad *
 sin(radians(rot) * cos(radians(tilt)), camRad *
 sin(radians(tilt)), -camRad * cos(radians(rot)) *
 cos(radians(tilt)))
    rottilt = False
```



› **Avete mai sognato di guidare un carro armato? Per farlo vi bastano poche righe di codice...**

Ora bisogna impostare i tasti da usare per controllare la posizione della videocamera; userete i tasti standard **W**, **A**, **S**, **D** per la rotazione e **+/-** per lo zoom. Modificate l'inizio del blocco che controlla la tastiera in questo modo:

```
 if k > -1:
   rottilt = True
   if k == 112:
     pi3d.screenshot("earth1.jpg")
   elif k == 119:
     tilt += 2.0
   elif k == 115:
    tilt -= 2.0
   elif k == 97:
     rot -=2
   elif k == 100:
     rot +=2
   elif k == 61:
     camRad -= 0.5
   elif k == 45:
     camRad += 0.5
```

Se volete saperne di più, trovate tutta la documentazione di questa libreria su **http://bit.ly/Pi3DDocs** e ci sono sempre tutte le altre demo da studiare a fondo. LXP

100% Wynn

Abbiamo parlato con Wynn Netherland dell'importanza delle API, dei collaudi e delle gioie di Ruby

Da quasi vent'anni, Wynn Netherland contribuisce a costruire Internet così come la conosciamo. Prolifico creatore e gestore di wrapper API scritti in Ruby, si occupa attualmente dell'API GitHub e ha pubblicato numerosi libri.

Linux Pro: **Quali sono i maggiori cambiamenti a cui hai assistito negli anni in cui ti sei effettivamente dedicato a costruire il Web?**

Wynn Netherland: Ottima domanda. Sotto molti aspetti è diventato più professionale, non c'è più quell'atmosfera tipo cowboy. Ricordo che all'inizio della mia carriera, presso Compaq e poi presso Hewlett-Packard, eravamo sostanzialmente dei pionieri nell'uso dell'Open Source per l'e-commerce, ancora a livello di ipotesi. In sostanza, si trattava ancora di cercare di escogitare un modo per accettare pagamenti online. All'epoca facevamo queste cose senza nemmeno RoboSource Control e standard analoghi, che oggi si danno per scontati. Allora era anche una questione di requisiti legali. Insomma, sotto molti aspetti il Web è cresciuto, è diventato più... aziendale. In un certo senso abbiamo perso un po' di... stavo per dire innocenza; di certo si è perso molto del divertimento, di quell'atmosfera tipo parco giochi in cui si viveva quando tutti cercavano di capire come fare determinate cose. Il Web non era ancora un insieme di giardini recintati... AOL a parte! Si direbbe che negli ultimi anni abbiamo sempre più rinunciato a un Web aperto, in cambio di una serie di giardini recintati. E per qualche ragione, come comunità lo accettiamo. Non saprei dire perché.



LXP: **Immagino che agli esordi, molti giovani iniziassero con un account Geocities che permetteva loro di creare un sito Web, farsi la mano con l'HTML e perfino provare a vendere qualcosa. Cominciare era relativamente semplice. Oggi molte persone, invece di imparare a programmare il proprio sito, si limitano a crearsi un blog Wordpress o una pagina Facebook.**

WN: È vero; e a quanto pare le pagine Facebook sono decisamente più standardizzate dei blog Wordpress. In un blog creato con Wordpress, con un altro servizio analogo o con strumenti CMS Open Source, anche se si inizia scegliendo un tema pronto, prima o poi ci si ritrova a maneggiare HTML e CSS, per modificarlo anche in parte. Con Facebook invece è tutto già pronto.

LXP: **Dunque Internet è cambiata, trasformandosi sempre più, come dici, in una serie di giardini recintati. A tuo giudizio ci sono stati anche dei cambiamenti positivi?**

WN: Come sempre quando sono coinvolti gli esseri umani, ci sono aspetti positivi e aspetti negativi. Proviamo a pensare all'interoperabilità: anche se i dati sono più segmentati, abbiamo una maggiore interoperabilità nel modo in cui scambiamo questi dati. Alla fine degli anni Novanta usavamo ancora strumenti come Corba; XML praticamente non esisteva ancora. Si inviava il codice HTML al browser e si sperava che lo renderizzasse correttamente! Oggi, invece, siamo abituati a strumenti che rendono semplicissimo lo scambio dei dati: JSON e HTTP sono l'equivalente Web dei toni della tastiera telefonica. A chi appartengono questi dati, però? In che modo posso esportarli e portarli con me? Questi sono alcuni dei principali problemi. E il maggiore di tutti probabilmente è: chi sta osservando i miei dati in questo momento?

LXP: **Sì, questa è una preoccupazione costante. Hai una passione dichiarata per l'esperienza utente delle API: perché è tanto importante?**

WN: Per quanto riguarda l'esperienza utente, ho iniziato la mia carriera nell'ufficio di progettazione di un giornale, dopo di che mi sono spostato sempre più a monte. Quando alcuni anni fa sono comparse le API aperte, ho cominciato a occuparmene per soddisfare una mia specifica esigenza, cioè estrarre i dati dai sistemi che utilizzavo. Volevo mettere insieme il mio blog con tutti i dati tratti dai punti che utilizzavo sul Web. Ero veramente frustrato dal fatto che la tendenza comune fosse quella di rendere quasi deliberatamente la vita difficile a chi tentava di estrarre quei dati! Credo che dipenda da una mancanza di empatia con l'utente. Quando si parla di esperienza utente, l'empatia è tutto. Oggi tendiamo a considerare l'interfaccia utente uguale all'interfaccia grafica, quasi sottintendendo la 'G' della sigla GUI (*Graphical User Interface*). È così che funziona Unix: testo che entra e testo che esce. Questa è un'interfaccia utente: in sostanza, non ha un elemento grafico. Per le API è lo stesso: anche adattando REST, è possibile fare riferimento a un archetipo di API che ci permette di metterci nei panni dell'utente, domandandoci per esempio: "Che cosa mi piacerebbe visualizzare?" o "Come rendere possibile un flusso di lavoro incentrato su questo strumento?", invece di limitarci a confezionare il nostro database in REST.

LXP: **Quindi, è chiaro che creare un'API stabile è straordinariamente importante: quanto importante, secondo te?**

WN: È della massima importanza. Direi che in un certo senso si tratta ancora di un obiettivo idealistico. Per il consumatore è facile dire: "Ehi, hai rovinato la mia app! Dovresti cambiare API!". All'interno della comunità che ruota intorno alle API esiste una minoranza piccola ma rumorosa secondo la quale creare più versioni di un'API è un errore in sé. Pensiamo a HyperMedia e simili. Certo, queste persone contribuiscono a mantenere limitati i cambiamenti; ma a meno di avere il dono della preveggenza, non credo sia sempre possibile farne a meno... A volte si »

» introducono cambiamenti perfino mentre si eliminano i bug e si fa in modo di mantenere l'API in linea con le soluzioni funzionanti già in circolazione. Immaginiamo che ci sia un bug in un'API pubblicata che funziona in un certo modo. A quel punto, ci sarà un intero gruppo di persone che avranno creato strumenti sulla base di quel bug. Se io mi accorgo del bug e lo elimino, mando all'aria gli strumenti creati da queste persone, che pure funzionavano.

LXP: Dunque occorre avere empatia per le persone che utilizzano l'API ma, al tempo stesso, c'è un momento in cui è necessario chiudere la porta e fare di testa propria: bisogna seguire la propria via, insomma?
WN: Esatto. Per quanto riguarda l'API GitHub, il nostro approccio parte da una quantificazione. Si tratta di misurare l'entità del cambiamento che introdurremo: refactoring, aggiunta o eliminazione di endpoint, questo genere di cose. Misuriamo il cambiamento, lo quantifichiamo. Sulla base di questi dati, cerchiamo di capire come ridurre al minimo il cambiamento. Questo ci permette di gestirlo. Perciò, se in un'API c'è un endpoint che nessuno utilizza, è possibile rimuoverlo senza danneggiare nessuno. In effetti, tutto si riduce all'impatto sugli esseri umani. Per esempio, possiamo eliminare un endpoint che risulta problematico per il team operativo e per l'infrastruttura o che mette a rischio un intero sito. Puntiamo sempre a ridurre al minimo i rischi e se ci troviamo costretti a eliminare un metodo di un'API, la prima cosa che facciamo è verificare chi lo utilizza e con quale frequenza, per poi metterci in contatto con queste persone e dire loro: "Cerchiamo dei sistemi alternativi per risolvere il problema".

LXP: Hai citato GitHub: come sei entrato in questo progetto?
WN: lAttraverso il wrapper dell'API. Alcuni anni fa stavo lavorando al podcast Changelog e una delle cose che volevamo fare era mettere in evidenza i numerosi sviluppi Open Source che caratterizzavano GitHub. Lo chiamavamo 'La storia di Changelog' e in sostanza si trattava di una vetrina di tutti gli sviluppi di GitHub in corso. Nessuno dei wrapper Ruby in circolazione all'epoca mi piaceva: aderivano troppo strettamente a REST nella mappatura e io sono fermamente convinto che i wrapper API debbano essere strettamente legati al linguaggio in cui sono scritti. In altre parole, se li si scrive in Ruby, dovrebbero essere simili a un'estensione di Ruby. Così ho creato un wrapper, che in seguito è divenuto octokit; e quando un paio di anni dopo mi è stato proposto di entrare nel progetto GitHub, mi sono reso conto che GitHub utilizzava quel wrapper internamente. Se lo avessi saputo, mi ci sarei impegnato di più! Per me quel progetto era stato un passatempo. La prima volta che ho usato Bootstrap sul codice di GitHub e l'ho visto installare octokit, ho pensato: "Wow!" Così, oggi lo utilizziamo come interfaccia API primaria per molti dei nostri strumenti, dato che buona parte di questi ultimi sono scritti in Ruby. Per esempio, viene utilizzato per il controllo delle autorizzazioni degli script interni, per le rilevazioni statistiche, per l'autenticazione di varie applicazioni interne. Tutte queste procedure utilizzano octokit internamente.

LXP: Devi esserti sentito orgoglioso vedendo utilizzare il tuo codice...
WN: Certo; ma ogni progetto Open Source è frutto di un lavoro di gruppo e nel corso degli anni vi hanno contribuito molte persone. Erik Michaels-Ober è la persona che probabilmente ha dato il maggior contributo a octokit. Lavora presso Soundcloud e ha realizzato numerosi wrapper. Quando però il team Mac ha voluto estrarre il suo wrapper API per Objective-C da GitHub per l'app Mac, aveva bisogno di un nome. Dare nomi alle cose e rendere invalide le cache: sono questi i grandi problemi dell'informatica, no? Era ovvio che il nome migliore per il client Objective-C era octokit, perciò ho detto loro: "Dovreste usare questo nome e dovremmo trasformarlo in una sorta di marchio collettivo, per così dire". Così il nostro progetto ha preso il nome di octokit, l'ho trasferito dal mio dominio personale di GitHub a **octokit.org** e ora ne abbiamo versioni per Objective-C e per .net. La gente continua a chiedermi perché non esiste una versione Python. In realtà ce n'è una in Go ma la nostra filosofia per quanto riguarda la pubblicazione di wrapper Octokit è che vogliamo che siano soluzioni che noi per primi utilizziamo, collaudiamo e manteniamo.
Non intendiamo mantenere uno strumento che non usiamo, man mano che diviene obsoleto.



LXP: Quali sono le tue principali responsabilità quale membro del team API di GitHub?
WN: Prima di tutto, far funzionare le cose essenziali. Al momento il team API è composto da due persone soltanto, Jason Rudolph e il sottoscritto. Siamo un piccolo team integrato in un team più grande, quello che definiamo '.com engineering team'. Perciò, la responsabilità principale consiste nel curare gli aspetti orizzontali dell'API in relazione all'intero sito: autenticazione, limitazione del traffico e tutti gli elementi utilizzati dalle analisi SWOT dell'API. Inoltre, forniamo indicazioni e suggerimenti ai team che si occupano delle funzioni, riguardo ai problemi legati alle pull request, all'accesso ai repository e all'estensione e alla manutenzione delle loro API, come estensione della loro

attività sul sito. L'API è una piccola applicazione integrata nell'applicazione Web più grande, perciò condividiamo gran parte del codice backend; solo i flussi di lavoro e l'output sono isolati.

LXP: Perciò effettuate anche gran parte dei test relativi al comportamento dell'API; in alcune conferenze hai parlato di alcuni degli strumenti da te utilizzati, come JSON. Puoi illustrarci la procedura usata per i test?

WN: Certo. Il principale formato di output da noi utilizzato per l'API è *JavaScript Object Notation* (JSON), perciò usiamo lo schema JSON. Uno dei suoi limiti è che a differenza di XML non è accompagnato da una tecnologia schema incorporata. È molto leggibile e fornisce dati specifici primitivi nel linguaggio prescelto ma non vi è modo di sapere quale formato dovrebbero avere i dati. Perciò utilizziamo lo schema JSON sia in input sia in output. Recentemente lo abbiamo utilizzato per l'input e stiamo facendo notevoli progressi. Tradizionalmente, in fase di output, quando avviamo il nostro sistema di collaudo, possiamo esaminare lo stream JSON ottenuto e convalidare per esempio che una data risposta di un oggetto abbia date proprietà. Ora stiamo iniziando a utilizzare questo sistema anche in fase di input, in modo da poter effettuare convalide. Perciò, quando viene creata una pull request, noi riceviamo uno stream JSON e possiamo sottoporlo a queste convalide, per poi dire per esempio: "Spiacenti, questo non è il formato giusto". Il sistema ci consente cioè di specificare nel test mediante dichiarazioni qual è l'aspetto di un documento valido. In questo modo non siamo costretti a scrivere test ad hoc per ogni permutazione per verificarne la validità.

LXP: Un altro strumento che utilizzate parecchio è Rack::Test...

WN: Rack::Test è valido per qualunque applicazione Rack: nel mondo di Ruby c'è un canale HTTP chiamato Rack. Alcuni anni fa, molti framework Web si sono uniti dando vita a un modello standardizzato di canale input-output chiamato Rack. Perciò oggi molti sono conformi a questa interfaccia che permette di creare una batteria di applicazioni intermedie, simile a un rack di server in un centro dati, per svolgere operazioni in formato pipeline. Se c'è un'applicazione che riceve una risposta HTTP, agisce di conseguenza e trasferisce la risposta all'applicazione successiva perché la utilizzi. Autenticazione, limitazione del traffico, gestione dei cookie, analisi dell'input o dell'output per la ricerca di vulnerabilità relative agli script tra più siti: parliamo di operazioni di questo genere. Usiamo Rack::Test per poter testare le nostre app quando è necessario.

LXP: È importante individuare ogni potenziale problema in questa fase. Siete in grado di tracciare con precisione l'uso che viene fatto della vostra API? Ritornando al tuo commento sull'empatia, è decisamente utile poter controllare come viene usata l'API e quali sono le sue prestazioni.

WN: Utilizziamo alcuni parametri di misurazione a più livelli. Uno riguarda le risposte in sé e il loro stato in corrispondenza di specifici endpoint. Conosciamo l'endpoint Events che è la nostra API utilizzata per le attività in streaming. Riceve un determinato volume e sappiamo che questi sono i tipi di applicazioni che le persone creano utilizzando GitHub come base. Quindi, uno dei livelli riguarda il volume in quanto tale; ma non sempre il volume fornisce di per sé informazioni sufficienti. Per il resto, dal momento che i metodi non hanno tutti lo stesso valore in termini di flusso di lavoro, direi che gli aspetti che mi entusiasmano davvero sono quelli in un certo senso più profondi, quelli che riguardano la creazione e la fusione di pull request e tutte le attività che si svolgono in un sito. Al di là dell'uso dell'API come semplice ambito per un mashup di dati da utilizzare come database, ciò che mi entusiasma è vedere applicazioni che su questa base creano interi flussi di lavoro, clonando un repository, aggiungendo funzioni e utilizzando l'API per inviare le pull request. Esistono molti strumenti che creano applicazioni per la revisione del codice sulla base di GitHub e rispecchiano ciò che facciamo con il sito ma si adattano meglio ai singoli flussi di lavoro.

WEB CONTRO NATIVO

"È una falsa dicotomia, perché anche le app native usano JSON e le API"



LXP: Un altro progetto Open Source da te citato nelle tue conferenze è Science. Puoi spiegarci di che si tratta?

WN: Il progetto nasce dalle idee di John Barnette e Rick Bradley di GitHub. Avevamo un problema specifico all'interno del codebase .com: per quanto questo sistema per i test sia eccellente (e sulla base dei progetti a cui ho lavorato, posso dire che è un sistema fenomenale), ci sono alcuni aspetti delle applicazioni, una sorta di 'ventre molle', per i quali non è possibile fare affidamento al 100% sui test. La ragione può essere che il framework utilizza tecniche di metaprogrammazione complesse, che costituiscono un linguaggio dinamico da mettere a punto nella fase di progettazione. Perciò, qualora vi sia un elemento fondamentale che non può essere modificato ma necessita di un refactoring, Science è un'applicazione, o meglio un progetto, che suggerisce soluzioni di programmazione alternative. Nel caso di un metodo read only, che non subisce mutazioni e restituisce un dato valore, come nel metodo scientifico, è possibile avviare entrambe le alternative di codice in parallelo, confrontarne i risultati e pubblicarli sotto forma di tabelle e grafici. Così è possibile confrontarle direttamente, dicendo: "OK, questo è il percorso di codice originale, quest'altro è il nuovo percorso. Differiscono nel 10% dei casi e dobbiamo capire dove ha origine questa variazione". Oppure il nuovo percorso potrebbe risultare allineato al 100% ma risultare anche più lento o più veloce. Science ci permette di operare questi confronti e di applicare le nostre soluzioni sulla base dei dati concreti, invece di limitarci a incrociare le dita sperando che funzionino. »

» **LXP:** **Immagino inoltre che vi sia utile per individuare gli schemi...**
WN: Sì, specie quando sono in gioco cose di sicurezza che non ammettono errori.

LXP: **Da quanto tempo usi Science?**
WN: Science è in circolazione... direi da un anno circa. Non è stato sempre Open Source. Ne ho parlato per la prima volta nel marzo 2014 e sta davvero prendendo piede nel team; un codebase invecchia rapidamente e stiamo cercando di capire come rendere le cose un po' più veloci e moderne sotto alcuni aspetti.

LXP: **Qual è l'importanza degli strumenti Open Source per la creazione e il collaudo delle API?**
WN: Direi che sono indispensabili. Diversamente, non saprei come sarebbe possibile operare al di fuori di un ecosistema Microsoft. La prima diapositiva che ho mostrato alla mia conferenza [all'OSCON dello scorso anno] riguardava il modo in cui intendo creare un'API resiliente in Open Source; ho iniziato a spiegare la questione al pubblico utilizzando strumenti Shell e tazze di tè... ho solo 40 minuti per la conferenza, perciò devo accelerare un po' i tempi! In realtà la questione riguarda tutto quanto, dai framework Web che vengono scelti, ai server sui quali funzioneranno, allo stack di strumenti che verranno utilizzati per le misurazioni e per raccogliere dati di ogni tipo relativi alle prestazioni del sito. Ognuno di questi aspetti implica una decisione: si tratta di capire quali strumenti utilizzare. Nella comunità Open Source c'è solo l'imbarazzo della scelta, perché si può sempre scegliere. È una sorta di tirannide della scelta; e spesso è gratificante, quando si entra a far parte di un progetto, scoprire che molte di queste decisioni sono già state prese, come è capitato a me con GitHub.

IL SOFTWARE STA DIVORANDO IL MONDO INTERO

"Stiamo arrivando a un punto in cui si rischia di sottovalutare la sua onnipresenza"

LXP: **Tornando quindi alle API, oggi esiste una grande quantità di dispositivi che possiamo utilizzare per accedere a Internet e ai servizi online. Questo ha reso più complesso lo sviluppo delle API?**
WN: Sicuramente ha sottoposto le API a un carico maggiore. L'altro giorno ho letto un'affermazione, che vale se non altro per il contesto dei dispositivi mobili, secondo cui la dicotomia 'Web contro nativo' è in un certo senso fasulla, dato che anche le app native utilizzano JSON e le API dietro le quinte. Nella maggior parte dei casi si tratta delle stesse applicazioni, che semplicemente vengono renderizzate in un formato diverso per i computer. Perciò, sicuramente è aumentato a dismisura il numero di applicazioni collegate al Web; non sono più solo i browser a farla da padroni. In un certo senso, però, è cambiato anche il nostro modo di concepire le app native. Ricordo quando ho realizzato il mio secondo o terzo wrapper API per un servizio oggi scomparso chiamato Gowalla [un social media a base locale]. All'epoca il suo rivale era Foursquare. Mentre in Foursquare gli utenti gareggiavano con tutti gli altri per totalizzare il punteggio maggiore in una certa località geografica, Gowalla somigliava più a una specie di zaino da viaggio personale. L'utente otteneva cioè un badge diverso per ogni stato degli Stati Uniti o per ogni nazione del mondo che visitava. Poteva inoltre lasciare oggetti virtuali ad altre persone. L'approccio era completamente diverso, più artistico e mi piaceva moltissimo. Ci lavoravano alcuni miei amici; io continuavo a incitarli a creare un'API e loro rispondevano che se ne stavano occupando ma tenevano la bocca cucita. Poi è uscita la loro app per cellulare e io ho scritto un wrapper API per Gowalla estraendo la loro API per cellulare, avviando la loro app tramite un server proxy. In questo modo ho scoperto molti aspetti entusiasmanti della loro API, frutto di decisioni intelligenti che oggi sono rispecchiate da altre API. Molte di queste soluzioni sono finite anche nell'API GitHub.

LXP: **Dev'essere piuttosto complicato effettuare i test quando si cerca di immaginare come i diversi utenti di Internet accederanno a determinati servizi o app...**
WN: I cambiamenti sono costosi e, come dicevamo prima, di norma non vengono accolti positivamente; perciò, ecco che cosa abbiamo fatto. Abbiamo iniziato a riflettere sugli endpoint delle versioni beta delle API; ciò che avviene normalmente è che qualcuno pubblica un'API e, una volta che questa ha raggiunto un minimo di diffusione, introduce una seconda versione dell'API stessa. Anche GitHub procedeva così in passato, passando da V2 a V3. Quando siamo passati a V3 abbiamo deciso di iniziare a creare versioni degli endpoint sulla base dell'output e non del percorso. In sostanza, veniva inviato uno specifico header HTTP, che richiedeva una versione specifica della risorsa in questione. Negli ultimi 18 mesi abbiamo iniziato a usare questo sistema come una sorta di sistema di password per le nuove funzioni. Pubblichiamo una versione beta di un'API e nello header compare la parola 'preview', che equivale a un contratto per cui l'utente accetta di non creare in fase di produzione nulla che faccia riferimento a questo endpoint dell'API ma, al tempo stesso, permette a noi di ottenere dei riscontri reali da parte degli utenti. È suscettibile di cambiamenti in qualunque momento e non diamo alcuna garanzia sulla durata di queste API in versione beta; ma provandoci un paio di volte abbiamo potuto ottenere un feedback estremamente utile da parte di utenti che hanno usato qualche particolare funzione in modi che non avevamo previsto. Abbiamo così apportato delle modifiche prima di pubblicare il prodotto finito e quest'ultimo è risultato molto migliore proprio grazie a queste modifiche.

LXP: **La partenza è stata davvero incoraggiante, insomma?**
WN: Certo; e non solo dal punto di vista



delle versioni beta. Ora, quando scopriamo che una particolare risorsa non funziona a dovere per una determinata categoria di utenti, possiamo offrire loro un diverso output media type più adeguato.
Per esempio, esistono campi che vanno rimossi quando si eliminano determinate funzioni da un'applicazione; e nei contesti Python e Ruby, eliminare un campo da JSON o impostarlo su null non rappresenta un grosso problema. Per i client statici non è altrettanto semplice, perché comporta problemi di tempo legati alla compilazione. Perciò, in questi casi possiamo offrire un media type diverso sull'endpoint in questione, mettendo a disposizione una versione diversa della risorsa, più adatta alle esigenze del caso.

LXP: Ruby è il linguaggio preferito nell'ambito di GitHub, giusto?
WN: Sì, direi che la grande maggioranza dei nostri strumenti utilizza Ruby. Ne realizzeremo sempre più in Go, per quanto riguarda il caricamento di materiali multimediali e simili, semplicemente perché è più adatto al caricamento in parallelo. Naturalmente, quando si tratta di creare un'app Mac occorre utilizzare Objective-C e ora Swift; i nostri team Mac stanno esaminando Swift e sono in procinto di adottarlo. E altrettanto naturalmente, il team che si occupa di Windows utilizzerà .net. Insomma, rimarremo sempre poliglotti; ma per quanto riguarda le app Web, la grande maggioranza è scritta in Ruby.

LXP: Che cos'ha Ruby di tanto utile?
WN: Beh, il motto 'ottimizzare per la felicità' è essenziale per la comunità di Ruby. Insomma, Ruby è... Prima di entrare a far parte della comunità di Ruby ero uno sviluppatore .net e per me personalmente non c'è niente di più gratificante che scrivere in Ruby. In alcuni casi ci si ritrova ad avere bisogno di prestazioni che Ruby non è in grado di fornire, essendo un linguaggio dinamico, sebbene sia possibile sfruttarlo veramente a fondo. Go mi piace molto ma nella maggior parte dei casi non è accessibile quanto Ruby. Questo è un aspetto interessante, perché ho scritto molto anche in JavaScript; ma la sintassi di JavaScript, con quel termine Function che compare dappertutto, in qualche modo, direi, si avvicina meno di Ruby al linguaggio umano... Le sue funzioni di metaprogrammazione permettono di ottenere risultati interessanti utilizzando linguaggi specifici... invece di scrivere decine di dichiarazioni, posso inserire tutto in cinque righe di DSL. Il risultato è molto chiaro, permette di dichiarare in modo esplicito che cosa deve avvenire, senza ricorrere allo stile verboso che ha caratterizzato in passato molti linguaggi di programmazione.

LXP: E poi è molto più accessibile ai neofiti. Non sono costretti ad aspettare fino alla fine per vedere il risultato...
WN: È molto facile da seguire, specie per chi muove i primi passi in Ruby... uno degli inconvenienti è che possono esistere tre modi diversi per fare una determinata cosa ed è possibile che l'utente non conosca la forma idiomatica da utilizzare in Ruby per fare una certa cosa. Insomma, la creazione può richiedere più tempo della fruizione ma il risultato è molto più facile da seguire.

IL WEB È CAMBIATO
"Negli ultimi anni siamo passati da un Web aperto a una serie di giardini recintati"

LXP: Recentemente GitHub ha annunciato di avere otto milioni di utenti, il che ne fa il maggiore host di codice del mondo. Qual è a tuo parere la ragione di questo successo?
WN: Il modello Open Source e il coinvolgimento in GitHub di sviluppatori e progetti di prestigio hanno avuto un ruolo eccezionale. Ricordo che agli esordi, quando ancora si parlava di versioni beta, per poter vendere a qualcuno GitHub occorreva prima vendergli Git. Lo stavo facendo durante un pranzo, e un signore seduto accanto a me era appena passato a Subversion con Git. L'intera tavolata è esplosa: "Oh, devi provare questo e quest'altro, a me è stato utilissimo". È stato uno di quei momenti in cui è come se si accendesse una lampadina: interi progetti che adottano GitHub, un'ondata di attività e di comunità che investe GitHub. L'aspetto più significativo è che non vi è più uno sbarramento per la maggior parte dei progetti. Nel repository Canonical tutto dipende da chi è più attivo: se il tuo progetto giace lì immobile, nulla impedisce alla comunità di creare un fork del progetto, che a quel punto diventa un progetto Canonical. Coloro che si occupano della manutenzione dei progetti devono realmente sforzarsi di tenere il passo con le comunità o di delegare. D'altronde siamo pur sempre esseri umani limitati, perciò con l'esplodere dell'attività si è sviluppato un nuovo modello: quando qualcuno presenta una pull request che appare ben fatta e solida, ciò che avviene (e che in passato sarebbe stato considerato inaudito) è che gli viene data via libera, in modo che possa contribuire a sviluppare il progetto, perché non è possibile mantenere vivi tutti i progetti Open Source; sotto questo aspetto, non è possibile tenere il passo.

LXP: Immagino che il passaparola all'interno della comunità Open Source risulti utile...
WN: Una cosa che continua ad appassionarmi è il fatto che se si esce dall'effetto eco prodotto dalle tipiche conferenze tecniche (in particolare nel campo della progettazione), relativamente a HTML5 e ad alcune altre tecnologie, l'intera comunità si interccia oggi con le comunità di sviluppo Open Source più tradizionali. Nelle conferenze più legate ad ambienti aziendali, la compenetrazione è molto minore ma si può notare che procede comunque con lo stesso ritmo. Per questo è molto incoraggiante immergersi in queste comunità e trascorrere qualche giorno con persone che hanno veramente vissuto quei momenti illuminanti in cui si pensa: "Ehi, questo potrebbe migliorare tutto quanto".

LXP: Che cosa riserva il futuro a GitHub?
WN: Si tratta di rivoluzionare il modo in cui le persone scrivono i software. Direi che all'interno di GitHub la parola d'ordine è che il software sta divorando il mondo. Basta riflettere sulla profondità con cui il software penetra in ogni aspetto della nostra vita. Si va dai dispositivi che teniamo nel palmo della mano agli impieghi meno divertenti, come quelli in ambito ospedaliero. Non ho paura delle iniezioni, ho paura del software... perché so com'è scritto! [Ride] D'altronde, il software è ovunque, no? E stiamo arrivando a un punto in cui si rischia facilmente di sottovalutare questa sua onnipresenza. Più software c'è intorno a noi, più gli strumenti dovranno evolversi per tenere il passo, praticamente in ogni settore. Io credo che GitHub sarà la base di partenza per questa attività. LXP





Fedora 21

Linux Pro vi porta alla scoperta dell'ultima versione di una delle distro più apprezzate

Abbiamo provato Fedora 20 circa un anno fa. Adesso, invece, è venuto il momento di mettere sotto i riflettori la versione 21. Prima di approfondire la conoscenza di questa distro, è importante ripercorrere per qualche minuto le varie vicissitudini che hanno portato al suo rilascio. Fedora 21, infatti, non è stata pubblicata in tempi brevi. Purtroppo ha dovuto subire diversi ritardi, a causa di bug e problemi non sempre risolti. Spesso, infatti, gli sviluppatori sono stati sul punto di pubblicare, ma si sono dovuti fermare per cause di forza maggiore. Adesso, però, i bug sono stati estirpati e il suo funzionamento generale è praticamente perfetto, tanto da essere disponibile per workstation, server e in versione cloud. La ISO, peraltro, può essere masterizzata tanto su DVD, quanto su una chiave USB capace di superare la dimensione della distribuzione. In questo caso parliamo di 1,4 GB. Fedora, come Debian, è sempre stata considerata una distro di livello medio-alto. Questo non perché sia difficile da utilizzare, quanto più per il dover subire una certa mancanza di interfacce grafiche. Il loro uso potrebbe rendere più semplice alcune attività che invece devono continuare a essere svolte tramite riga di comando. Un fattore, questo, che in parte ne condiziona l'utilizzo.

> **"Fedora soffre della mancanza di GUI che consentirebbero un uso più facile di alcune opzioni"**

Potremmo dedicare gran parte di questo articolo a elencare nel dettaglio tutti gli aggiornamenti di versione e i pacchetti necessari a installare Fedora. Tuttavia non è certo questa la nostra intenzione. Infatti, per la maggior parte degli utenti tutto filerà liscio senza alcun problema. L'unico eventuale inconveniente che potreste affrontare riguarda l'installazione di driver proprietari. Tuttavia, in questo senso non c'è alcuna soluzione che possa migliorare le cose. Dovrete quindi armarvi di pazienza e tentare di risolvere per le buone. Volendo iniziare l'analisi di questa distro dal suo desktop, possiamo affermare che si tratta di una delle poche che sfrutta Gnome Shell di default. Se però non ne siete amanti o non avete dimestichezza con questo ambiente, non c'è assolutamente niente di cui preoccuparsi. In giro ci sono numerosi altri desktop che possono essere utilizzati al posto di Gnome. La procedura per la sostituzione è semplice. Basta installarli, quindi accedere alla schermata di log-in e scegliere l'ambiente che si preferisce.



› **L'applicazione per il meteo è molto pratica e comoda da utilizzare. Basta avviarla e scegliere la città di cui si vogliono visualizzare le previsioni**

Tre è il numero magico

Proseguiamo la nostra analisi puntando sugli elementi degni di nota. In primo luogo questa versione di Fedora non può essere considerata una sola distro, bensì tre. Infatti è stata suddivisa per piattaforme: cloud, server e workstation. Il lavoro maggiore svolto dagli sviluppatori è stato rendere l'immagine per la cloud meno ingombrante possibile. L'intervento chirurgico, in particolare, si è svolto sul modulo kernel. Questo è adesso un meta-pacchetto, a cui possono essere aggiunte altre funzioni andando a pescare da un ulteriore pacchetto chiamato **kernel-modules**. Il **kernel-core** contiene, oltre al kernel in sé, solo quei moduli richiesti per lavorare (OpenStack o EC2). La versione cloud di Fedora, quindi, si basa solo sul pacchetto kernel-core, mentre quelle dedicate alle workstation e ai server dipendono dal meta-pacchetto che contiene sia mudules sia core. Fedora Cloud dispone anche di un'altra funzione: **Project Atomic** di Red Hat. Questo consente di impostare un così detto Atomic Host, una piattaforma molto leggera costruita con CentOS e Red Hat Enterprise Linux, il cui unico scopo è fornire un host per contenitori Docker. Si tratta senza dubbio di un progetto ambizioso e in diretta concorrenza con la distro CentOS corrente. Project Atomic, peraltro, porta con sé l'eliminazione totale della tradizionale gestione dei pacchetti. Il sistema operativo, infatti, viene aggiornato con una nuova tecnologia chiamata OSTree. Il suo funzionamento è simile a un repository Git che ha il vantaggio di consentire aggiornamenti incrementali e ristabilire le funzioni principali in caso di problemi. Sul lato server, Fedora porta con sé diverse nuove tecnologie per semplificare il provisioning e l'amministrazione di sistema. Rolekit fornisce un sistema di distribuzione ottimale, tanto che i server possono essere equipaggiati rapidamente per eseguire una determinata funzione o uno specifico ruolo. Ognuno di questi può poi essere lanciato tramite la stessa interfaccia, migliorando la semplicità e la versatilità d'uso. Il centro di controllo è rappresentato da **Cockpit**, un'interfaccia Web-based che fornisce tutti gli strumenti più utili cui mettere mano. Una soluzione, questa, che attrarrà soprattutto i neofiti dell'amministrazione Linux. Proprio come il pannello di controllo Webmin, Cockpit consente di avviare e interrompere i servizi, nonché controllare i file di log ed eseguire qualsiasi tipo di manutenzione ordinaria e straordinaria. Cockpit, inoltre, può anche gestire server remoti. La caratteristica più ambiziosa di Fedora 21 Server è l'introduzione di **OpenLMI**, un'infrastruttura di gestione Linux che si propone di astrarre gli strumenti **myriad**, **syntaxes** e **idiosyncrasies** che i Sysadmind hanno fin'ora dovuto imparare. OpenLMI interviene con una API standard accessibile da C/C++, Python, Java o altri strumenti a riga di comando vecchio stile che permettono operazioni di routine da »

Aggiornamenti

Se state già utilizzando Fedora e volete aggiornare la vostra distro in modo rapido e semplice, il sistema per la selezione dei pacchetti può darvi una mano. È possibile utilizzare lo strumento **fedup** che potete installare con i seguenti comandi:

```
$ sudo yum update
$ sudo yum install fedup
```

Seppure sia possibile utilizzare una ISO come fonte di aggiornamento per fedup (usate l'opzione **--iso**), è più semplice filtrare il tutto direttamente da Internet. Potete farlo tramite queste istruzioni:

```
$ sudo fedup-cli --network 21
--product=nonproduct
```

In tal modo riuscirete a estrarre i pacchetti necessari. Basterà quindi riavviare il sistema e procedere all'installazione. Se volete che la vostra distro venga aggiornata con tutti i componenti di una versione specifica, per esempio quella dedicata alle workstation, utilizzate il comando **--product=workstation**. Ricordate però che questo sovrainstallerà Gnome su qualsiasi desktop stiate utilizzando.

» svolgere con facilità. OpenLMI fornisce anche un'interfaccia client-server e quindi si presta molto bene alla gestione e al monitoraggio remoto. A questo punto può quindi sorgere il dubbio che gli strumenti messi a disposizione da OpenLMI e Cockpit si sovrappongano, andando a creare una specie di doppione. In questi termini non ci sono pericoli. Infatti, entrambi lavorano su binari differenti e non si calpestano i piedi a vicenda. OpenLMI è in grado di utilizzare il tradizionale *Distributed Management Task Force* per gestire i vari dati e in più riesce a lavorare in coppia (anziché sostituire) con i programmi di sistema standard. Chi non ha mai utilizzato OpenLMI potrebbe trovarsi sopraffatto dalle nozioni da imparare. Tuttavia, la curva di apprendimento non è molto ripida e basta poco per prendere confidenza con gli strumenti a disposizione. L'ultima versione di Fedora cui non abbiamo ancora accennato è la workstation che purtroppo non porta con sé grandi novità. Il processo d'installazione è semplice e si sviluppa nelle solite schermate passo a passo. Il filesystem di default è ancora **Ext4** (probabilmente lo rimarrà fino a Fedora 23), ma **Btrfs** può essere utilizzato senza problemi e con la massima semplicità. Se si usano le impostazioni predefinite, vi ritroverete con tre partizioni: avvio, root e swap, dove le ultime due sono gestite tramite LVM.

Un aiuto agli sviluppatori

Il desktop predefinito è **Gnome 3.14** che viene fornito con un sognante sfondo blu. Colore, questo, che per anni è stato il simbolo di Fedora. Va detto che Gnome 3.15 è stato rilasciato a novembre e seppure non ci sia stato modo di installare la versione aggiornata, esistono sistemi per procedere dopo l'installazione. Per quanto riguarda la presenza di altre applicazioni, trovate LibreOffice 4.3, Firefox 33, Shotwell, Rhytmbox, Transmission e molti altri. Forse meno conosciuto è **DevAssistant**, un progetto Fedora iniziato due anni fa. Il suo compito è rendere la vita dei programmatori molto più facile. Infatti, consente di gestire ambienti di sviluppo, dipendenze e pubblicazioni del codice in modo molto più semplice e veloce. Così facendo, gli sviluppatori possono concentrarsi sulla scrittura, anziché su ciò che riveste un ruolo secondario. Naturalmente non mancano alcune applicazioni principali di Gnome, tra cui **Documents** che ricordiamo è basata su cloud. Ci sono poi **Google Maps**, una sorta di clone di Instagram/Snapchat chiamato **Cheese**, **Brasero** e un comodo programma per le informazioni meteorologiche. Gnome, inoltre, supporta anche lo streaming multimediale certificato DLNA, la condivisione di file via WebDAV e la condivisione dello schermo tramite il client VNC Vino. Circa il 50% delle applicazioni Fedora richiede l'uso dei file AppData necessari alla corretta visualizzazione dei programmi. A questo proposito, Red Hat ha incoraggiato i propri sviluppatori a fornire proprio questo genere di strumenti (Gnome Software, adesso completamente ridisegnato, non è più appannaggio dei soli utenti Fedora, ma viene incluso anche in Arch, Debian e OpenSUSE). Fedora, inoltre, utilizza anche Gnome nella sessione di Wayland. Ovviamente, non tutte le applicazioni supportano quest'ultimo desktop, ma in tal caso è possibile sfruttare l'alto grado di compatibilità di Xwayland. Wayland, lo ricordiamo, sarà il futuro rispetto a X e fornirà nuove e più potenti funzioni. Per ora, sebbene il suo stadio di sviluppo sia avanzato, non è ancora pronto per succedere al predecessore, quindi dovrete

› **Installare Plasma 5 è un gioco da ragazzi con *DNF*. Dovete solo scaricare 218 pacchetti...**

Come rendere unica Fedora

Le tre versioni di Fedora sono state create appositamente per un certo carico di lavoro, così come per esigenze particolari. Certo, è possibile personalizzarle, ma in ogni caso non saremo in grado di stravolgerle. In tal senso, si potrebbero provare KDE Spin che utilizzano Plasma 4 Desktop o una qualsiasi delle altre Spin desktop che comprendono Xfce, XLDE e Mate-Compix. Oltre ai desktop, ci sono Spin personalizzate per scienziati, appassionati di robotica, giocatori e molto altro ancora. Nel caso, si possono fare anche dei mix molto particolari e originali, tanto da sottoporli a Fedora per l'eventuale approvazione. Il procedimento per installare le Spin è ben documentato all'indirizzo **https://fedoraproject.org/wiki/Spins_Process**. L'idea generale è comunque partire da una configurazione kickstart preconfigurata che possa provenire da una release o una Spin ufficiale. Da qui si potrà poi aggiungere o sottrarre quello che si desidera. Dopo aver ordinato tutto in modo che sia conforme alle linee guida di Fedora, è possibile consultare Spin Wrangler che deciderà se la vostra Spin è completa e precisa. Da qui, se tutto risulta a posto, si procederà verso lo Spins SIG (*Special Interest Group*) che valuterà se il vostro lavoro è degno di passare al vaglio della Board per l'approvazione finale. Certo, a vederla così sembra una trafila burocratica niente male, ma non si può certo lasciare che il marchio Fedora venga traviato da chiunque.

accontentarvi di un semplice assaggio che tuttavia ne mostra le potenzialità in modo inequivocabile. Siamo quindi giunti all'analisi del gestore pacchetti. **Yum**, per certi aspetti, può essere paragonato ad Apt, il quale fa derivare alcune sue caratteristiche da Debian. Yum aggiorna automaticamente un database SQLite locale con i pacchetti disponibili e questo significa che non c'è alcun bisogno di comandi di aggiornamento separati. In altre parole, sarà impossibile trovarsi in una situazione in cui il sistema tenta di installare un pacchetto più vecchio di quello disponibile in un repository. Un'altra caratteristica degna di nota è il suo uso di delta-RPM, il quale, sfruttando l'utility **diff**, può farvi risparmiare parecchia banda in download. Yum è scritto in Python ed è estensibile tramite un sistema di plug-in. Per avere un elenco aggiornato di tutti i componenti aggiuntivi, usate il comando:

```
$ yum search yum-plugin
```

In questo caso, vi verranno mostrati soli quelli ufficiali, ma tenete conto che in Rete ce ne sono altrettanti unofficial.

Nuovo gestore pacchetti

Un'altra caratteristica molto importante di Yum è che in Fedora 22 sarà sostituto da **DNF** (alcuni sostengono che si tratti dell'abbreviazione di dandy Yum, ma la risposta ufficiale non è ancora arrivata). Comunque non c'è da preoccuparsi, perché la sintassi è molto simile. In più si può installare il pacchetto **DNF-yum** per fornire un wrapper per DNF. Il suo sviluppo è stato portato avanti fin da Fedora 18, ma solo nella versione 20 è stato ritenuto pronto per l'uso quotidiano. DNF può essere considerato una fork di Yum, con una base di codice molto più ordinato grazie soprattutto alle funzionalità di una libreria di backend chiamata Hawkey. I punti di forza di questo gestore pacchetti possono comunque essere ricapitolati come segue.

» **Estensibilità** Mentre i plug-in possono essere scritti per Yum, l'API non è ben documentata e tutto può essere fatto in Python. DNF, o più correttamente Hawkey, fornisce un' API pulita cui si può accedere tramite C o Python. Questo porta con sé un notevole miglioramento nell'accesso con i front-end come Gnome Software.

» **Risoluzione delle dipendenze** Yum, come sappiamo, non è indenne dai problemi di dipendenze irrisolvibili. DNF, invece, utilizza libsolv di OpenSUSE che riduce di molto i vari inconvenienti di questo genere.

» **Velocità** Dal punto di vista dell'utente finale, DNF permette anche di sincronizzare i metadati in modo simile a quanto farebbe un servizio in background tramite un cron job. Questo significa che in primo luogo potreste vedere dei picchi casuali di traffico Internet. In seconda battuta, invece, l'installazione dei nuovi pacchetti e il conseguente aggiornamento di sistema sarà caratterizzato da circa metà del tempo rispetto a prima. DNF rende poi ancora più facile l'utilizzo dei nuovi repository COPR.

> **"DevAssistant rende la vita degli sviluppatori molto più facile"**



› ***Boxes*** **consente di creare macchine virtuali in un baleno. Basta seguire le istruzioni passo a passo e in men che non si dica avrete la vostra virtual machine pronta all'uso**



› ***Gnome Software*** **colpisce fin da subito per il nuovo design mutuato in certi aspetti dall'Ubuntu Software Center. Ciò nonostante è pratico e molto facile da usare**

Si tratta di repo analoghi ai PPA di Ubuntu che contribuiscono a snellire il processo di confezionamento del software in un repository di facile accesso. Prima di COPR, infatti, per utilizzare i pacchetti di Fedora non ufficiali era necessario utilizzare Koji Build System o OpenSUSE Build Service. Queste due soluzioni, essendo sistemi completi, sono anche molto pesanti. Inoltre, dal momento che Koji distribuisce anche i pacchetti ufficiali di Fedora, chiunque lo utilizzi deve conformarsi alle sue linee guida piuttosto pedanti. Pertanto, analizzando tutti questi fattori, possiamo dire che COPR non è un sostituto di Koji, bensì il suo contrario. Infatti, si tratta di un sistema di compilazione leggero, facile da usare e ricco di caratteristiche. Per esempio, se si vuole eseguire Plasma 5.1, l'ultimo desktop di KDE, basterà usare i seguenti comandi:

```
$ sudo dnf copr enable dvratil/plasma-5
$ sudo dnf install plasma5
```

È possibile poi installare la suite di applicazioni KDE 5 con l'aggiunta del repository **kde-applications-5** con lo stesso metodo già visto sopra. Va poi ricordato che sebbene le librerie KDE 5 Framework siano già presenti nei repo standard di Fedora, Plasma e le applicazioni di cui abbiamo appena parlato non sono ancora considerate stabili. Per questo motivo, quindi, devono essere gestite diversamente. Per riassumere quanto abbiamo esposto in questo articolo, Fedora 21 è una distribuzione che si divide in tre ed è in grado di assumere l'onere di una grande eredità passata. Il nuovo approccio dimostrato con questa distro segna un passo importante e che vede ogni release come un nucleo ottimizzato per l'uso che se ne fa. LXP



Premiata Amministeria Dott. Brown

Dott. Chris Brown
Si occupa di formazione, scrittura di articoli e consulenze su Linux. Trova che il suo PhD in fisica delle particelle non sia di aiuto in questo lavoro...

Tecniche esoteriche per i sysadmin dai recessi più impenetrabili della sala server

Dattilografia

Una delle cose che trovo stupefacenti a proposito dei professionisti del computer che incontro è l'esigua percentuale di essi che ha davvero imparato la dattilografia. Passano la maggior parte della giornata interagendo con i computer attraverso una tastiera, ma molti di loro non sanno far altro che picchiettare con due dita, scorrendo laboriosamente la tastiera con gli occhi per trovare i caratteri di cui hanno bisogno. Anche la loro accuratezza lascia spesso a desiderare, tanto che a volte mi capita di pensare con preoccupazione a quando si trovano a usare una shell di root su un server di produzione. Ho appena misurato la mia velocità di battitura e risulta essere di circa 50 parole al minuto quando preparo un testo. (Nel gergo della dattilografia una "parola" è composta da cinque battute). Un dattilografo professionale medio è in grado di batterne 50-100. Io sono molto più lento, circa 30-40 parole per minuto, quando copio da un testo stampato. Ma c'è una enorme differenza tra l'essere in grado di battere della prosa e la battitura di strani comandi shell o codice di programmazione, in cui ci si trova a lottare con ogni sorta di caratteri non alfanumerici, lettere maiuscole nel mezzo di nomi di variabili e in cui bisogna tenere conto dell'indentazione e così via. Per esempio io riesco a inserire solo 20 parole al minuto quando scrivo codice Perl. E poi bisogna notare come le potenziali conseguenze di un errore siano molto più gravi. Pensate alla differenza tra lo scrivere:

```
$ rm *.old
```

e:

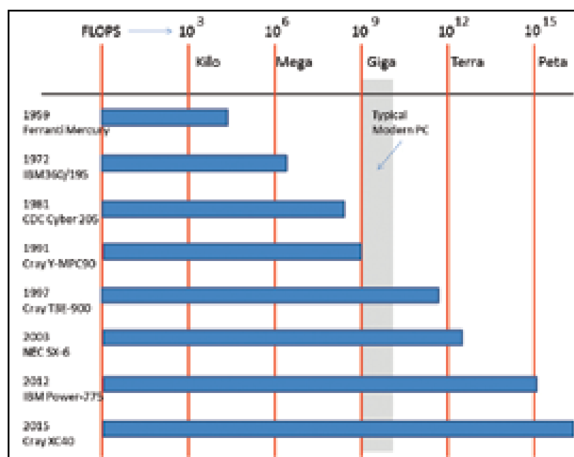
```
$ rm * .old
```

E voi, quanto veloce battete sulla tastiera? Pensate che una ridotta velocità di battitura o un scarsa accuratezza limitino significativamente la vostra produttività come amministratori di sistema? Oppure il tempo passato a pensare *cosa* dovete digitare è predominante sul tempo effettivamente speso per digitarlo?

Cray scalza IBM per le previsioni meteo

Un nuovo significato al termine *cloud computing*: il servizio meteorologico britannico punta su Cray

Il **Met Office**, il servizio meteorologico nazionale del Regno Unito, sta investendo più di 100 milioni di euro per acquistare un nuovo supercomputer, un Cray XC40. Quasi non serve aggiungere che ci girerà Linux. Secondo il sito Web di Cray il "Cray Linux Environment... comprende un sistema operativo basato su Linux e progettato per eseguire applicazioni grandi e complicate e in grado di scalare efficientemente fino a più di 500.000 core. Il Cray Linux Environment è dotato di un kernel che può essere configurato per adattarsi a diversi carichi di lavoro". Potete trovare i dettagli su **http://bit.ly/CraySpecs**. Il Met Office non è l'ultimo arrivato in fatto di supercomputer: acquistò la sua prima macchina (un Ferranti Mercury) nel 1959. Era in grado di eseguire 30.000 elaborazioni al secondo. Per fare un confronto, un PC moderno è forse 100.000 volte più veloce. Ma da allora ha acquistato non meno di sette supercomputer da IBM, CDC, Cray e NEC. La nuova macchina, che dovrebbe essere operativa a partire da settembre, avrà 480.000 core (dei Pentium Xeon) e ha una capacità di calcolo di 16 petaflop. Ho chiesto a mia moglie cosa sia un petaflop e lei ha risposto che pensa che si tratti di quel periodo della tarda estate in cui i fiori cominciano ad appassire: in realtà peta indica 10^{15}, un numero così incredibilmente grande che si rischia il mal di testa al solo pensarlo. Ho una sufficiente esperienza nell'utilizzo di macchine parallele per poter affermare che ben difficilmente riusciranno anche solo ad avvicinarsi a un simile valore in termini di calcolo. Si tratta comunque di un numero molto grande, circa 13 volte più potente dell'IBM Power 775 che attualmente ha il Met Office. Secondo la mia interpretazione della lista dei computer più potenti del mondo (**http://www.top500.org/lists**) dovrebbe trovarsi circa al quarto posto al momento in cui scrivo. Come si giustifica il costo di una macchina così potente? Beh, il Met Office afferma che produrrà 2 miliardi di euro di "benefici socio-economici" per il Regno Unito grazie a una "aumentata resistenza al maltempo e ai rischi collegati" ottenuta fornendo previsioni più frequenti e che permetterà ad aree strategicamente importanti, come gli aeroporti, di ricevere previsioni di velocità del vento, nebbia e neve con una risoluzione spaziale di 300 metri.



› **Negli ultimi 50 anni il Met Office ha visto una crescita esponenziale della sua potenza di calcolo**

Configurare un server OpenLDAP

Il Dottore vi mostra come centralizzare le informazioni sugli account degli utenti configurando un server OpenLDAP su Ubuntu

La lettura di questa mattina è presa dal libro di Tux, capitolo quinto, versetto primo. In principio era il file delle password, e il file delle password era con Unix. Per mezzo di esso tutti gli utenti erano collegati; senza, nessuno di quelli che si erano scollegati poteva entrare. E gli amministratori di sistema videro che era una cosa buona. Ma poi venne il tempo della Grande Rete e gli amministratori parlarono tra di loro dicendo "Il file delle password non funziona bene, perché richiede repliche di dati e non si adatta a grandi reti." E Sun disse "Non temete, perché vi darò le Yellow Page affinché possiate centralizzare i vostri dati". Ma poi arrivarono dei saggi e dissero: "Non puoi prendere il nome di Yellow Page perché è stato registrato da noi come marchio commerciale". E Sun disse "Da oggi in poi quello che era chiamato Yellow Page sarà chiamato NIS". E gli amministratori di sistema videro che era una cosa buona. Ma dopo poco tra gli amministratori sorse di nuovo il malumore e si lamentarono per la seconda volta dicendo: "In verità NIS possiede solo uno spazio dei nomi non gerarchico e nessun controllo degli accessi". E di nuovo Sun disse: "Non temete, vi darò NIS+ che avrà uno spazio dei nomi gerarchico e controllo degli accessi in abbondanza". Ma gli amministratori di sistema si lamentarono una terza volta, perché non comprendevano. E così fu creato un grande consorzio affinché definisse le specifiche X.500. E X.500 generò DAP, e DAP generò DIXIE, e DIXIE generò LDAP. E gli amministratori di sistema videro che era cosa buona. Qui si conclude la lettura di questa mattina. Questo mese (abbandonando rapidamente il mio vernacolo scritturale) impareremo le basi di **LDAP** e vedremo come configurare un servizio LDAP di directory per conservare gli account degli utenti. Il mese prossimo vedremo, tra l'altro, come configurare una macchina per usare un server LDAP come sorgente di informazione sugli account.

Un'introduzione a LDAP (solo un'infarinatura)

LDAP sta per *Lightweight Directory Access Protocol*, ma in generale quando parliamo di LDAP intendiamo anche riferirci al server che effettivamente parla il protocollo e conserva le informazioni nella directory. In linea di principio è possibile conservare qualsiasi tipo di informazioni in LDAP, ma in pratica viene di solito usato come una sorta di agenda degli indirizzi aziendali, in cui vengono conservati nomi utente, numeri di telefono, indirizzi postali, indirizzi di posta elettronica, ruoli all'interno dell'azienda e delle sue divisioni e così via. In particolare LDAP è in grado di archiviare le informazioni relative agli account degli utenti: quel genere di cose che veniva tradizionalmente scritto in **/etc/passwd** e in **/etc/shadow**. Una directory LDAP conserva le informazioni in una struttura ad albero molto simile a quella di un filesystem (o a quella del DNS). Questo albero è chiamato **DIT** (*Directory Information Tree*). Ogni elemento all'interno dell'albero è identificato da un *distiguished name*, qualcosa come **uid=mary,ou=People,dc=example,dc=com**. La prima parte (**uid=mary**) è chiamata *relative distinguished name*, mentre il resto è il *distinguished name* del nodo genitore (**ou=People,dc=example,dc=com**). È qualcosa di vagamente simile a un percorso completo all'interno di un file system Linux, come **/home/chris/articles/ldap**, in cui **ldap** è il nome del file, mentre **/home/chris/articles** è il percorso della directory genitore. Notate però come i componenti siano elencati in ordine inverso: i distinguished name sono scritti *little-endian* e i percorsi *big-endian* (i nomi DNS sono anch'essi scritti *little-endian*).
Il distinguished name dell'elemento più in alto nella directory (**dc=example,dc=com** nel nostro esempio) è chiamato *naming context* della directory e di solito viene ricavato dal nome DNS dell'organizzazione a cui si riferisce (**example.com**) perché questo è garantito essere unico. Assegnare al naming context il solo valore **dc=com** non è corretto perché la nostra directory non è pensata per contenere le informazioni relative all'intero dominio .com! Ogni elemento nella directory è sostanzialmente una raccolta di attributi e valori. In breve creeremo un elemento per un utente chiamato **mary** che comprende (tra molti altri) gli attribuiti di:

```
uid: mary
sn: Brown
givenName: Mary
```

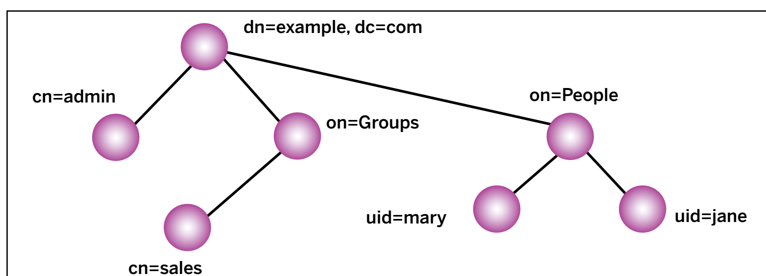
Gli attributi sono più o meno come le variabili nei linguaggi di programmazione, in cui possiamo dire che **uid** è una variabile con il valore di mary. Ma non spingete troppo in là questa analogia, perché, a differenza delle variabili, gli attributi possono contenere più valori. Per esempio per mary potremmo vedere quanto segue:

```
telephoneNumber: 01263 987654
telephoneNumber: 07639 123456
```

perché nella vita reale Mary ha due numeri di telefono. Ogni elemento nella directory deve avere un attributo chiamato **objectClass**. Il suo valore specifica una sorta di *template* che a sua volta indica quali altri attributi devono essere presenti nell'elemento e quali invece sono opzionali. Quindi, pensando a qualcosa di più completo per l'utente mary, potremmo vedere qualcosa di questo genere:

```
dn: uid=mary,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
```

»



› L'*LDAP Directory Information Tree* sviluppato in questo tutorial

```
objectClass: shadowAccount
uid: mary
sn: Brown
givenName: Mary
```

Come avrete notato l'**ObjectClass** di mary ha tre valori: *inetOrgPerson*, *posixAccount* e *shadowAccount*. Questi tre valori vengono combinati tra loro per determinare quali attributi debba avere l'elemento della directory che si riferisce a mary. I programmatori che conoscono la tecnologia orientata agli oggetti possono confrontare queste definizioni di **objectClass** con le classi, ed effettivamente una object class può ereditare da altre, per cui risulta che **inetOrgPerson** eredita da **organizationalPerson** che a sua volta eredita da **person**. Queste definizioni di classe costituiscono quello che è noto come *schema* della directory. Ma vi sto probabilmente spaventando con tutta questa teoria. In pratica non c'è bisogno di approfondire molto se si vuole semplicemente usare LDAP per centralizzare gli account degli utenti. Quindi procediamo...

Configurazione di LDAP

La nostra missione questo mese consiste semplicemente nel configurare un server LDAP per conservare le informazioni sugli account degli utenti. Userò Ubuntu 14.04 per fare questo. I principi sono i medesimi per qualsiasi distribuzione Linux, ma i dettagli possono essere diversi. Ho già trattato l'argomento in un articolo precedente (**LXP 147**) in cui ho mostrato come usare **Winbid** per accedere alle informazioni su un account all'interno di una Active Directory, che naturalmente si basa su LDAP. Questa volta però vogliamo implementare un nostro servizio LDAP. Stiamo per installare il server OpenLDAP, chiamato **slapd**, che si configura automaticamente a partire da un minimo di informazioni fornite dall'utente. Ha bisogno però di definire il suo naming context (il nome dell'elemento di livello massimo della directory) a partire dal nome della macchina, quindi conviene cominciare assicurandosi che sia inserito nel file **/etc/hosts**, qualcosa di questo genere:

```
127.0.0.1 localhost.example.com localhost
27.0.1.1 chris-hp250.example.com chris-hp250
```

Ora possiamo procedere con l'installazione dei pacchetti:

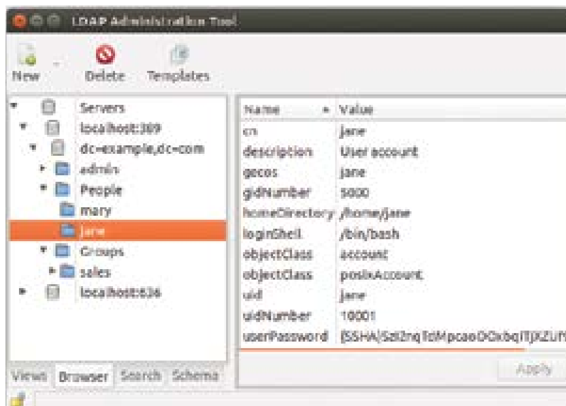
```
$ sudo apt-get install ldap-utils
```

Il comando installerà una mezza dozzina di strumenti lato client e le relative pagine di manuale. Installiamo ora il server:

```
$ sudo apt-get install slapd
```

Vi verrà chiesta la password per l'account amministratore LDAP, che è **cn=admin,dc=example,dc=com**.

Il pacchetto **slapd** contiene il server LDAP principale e un certo numero di strumenti di supporto con le relative pagine di manuale, una collezione di schemi e un certo numero di librerie di supporto. Normalmente vi aspettereste che un server abbia un file di configurazione, magari **/etc/slapd.conf**. In effetti le prime versioni di OpenLDAP lo avevano, ma ora le informazioni di configurazione sono state spostate nel suo DIT. I file LDIF da cui viene caricato il DIT sono conservati nella directory **/etc/ldap/slapd.d/cn=config**. Questi file *non* vanno assolutamente modificati manualmente. Se avete bisogno di dettagli date un'occhiata a **http://bit.ly/OpenLDAPAdminGuide**.



› ***LAT** (**LDAP Administration Tool**)* **è uno dei tanti strumenti grafici che permettono di navigare, fare ricerche e gestire i dati contenuti in una directory LDAP**

Creare un utente

Come accade di solito nelle distribuzioni derivate da Debian, l'installazione di un servizio automaticamente ne configura il server e lo porta in uno stato di funzionamento minimale. Possiamo quindi iniziare ad aggiungere dei contenuti. Cominciamo con la maniera più difficile per farlo. Per prima cosa create un file chiamato **populate.ldif** fatto così:

```
dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
ou: People
dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups
dn: cn=sales,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: sales
gidNumber: 5000
dn: uid=mary,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
```

Directory e database

Sia le directory come LDAP che i database come *MySQL* offrono una modalità altamente strutturata di archiviazione e recupero dei dati. Sono però molto diversi. Per prima cosa i dati in LDAP esistono all'interno di una struttura ad albero, cioè gerarchica. Non c'è alcuna possibilità di avere una qualche forma di "connessione" tra i differenti rami dell'albero. I database, invece, conservano le informazioni in tabelle e possono rappresentare relazioni del tipo chiave esterna/chiave primaria tra queste tabelle. È vero che uno schema LDAP (che definisce i tipi degli attributi che possono comparire all'interno di un nodo) è analogo allo schema di una tabella di database, ma non c'è alcun modo per eseguire un *join* tra due pezzi di una directory LDAP. Un altro fattore caratterizzante è che le directory sono progettate per essere "per lo più lette". Di solito lo sforzo richiesto per aggiornare un elemento in una directory è superiore a quello richiesto per recuperarlo. Come caso estremo c'è l'aggiunta di un singolo utente in NIS, che richiede la completa ricostruzione della mappa delle password. Ancora più estremo è il caso delle guide del telefono stampate, che vengono consultate ogni giorno da migliaia di utenti. L'aggiornamento di una directory di questo genere richiede la stampa e la spedizione di un sacco di carta ed è fatto una volta all'anno. Con i DB gli aggiornamenti sono più frequenti e il rapporto lettura/scrittura è più equilibrato.

```
uid: mary
sn: Brown
givenName: Mary
cn: Mary Brown
displayName: Mary Brown
uidNumber: 10000
gidNumber: 5000
userPassword: marybrown
gecos: Mary Brown
loginShell: /bin/bash
homeDirectory: /home/mary
```

Si tratta di un esempio di un file LDIF, un formato che usa semplici file di testo per dare una rappresentazione "esterna" del contenuto di una directory LDAP. Questo file descrive due importanti elementi di livello massimo: **People** e **Groups**. Al di sotto di essi abbiamo aggiunto un gruppo chiamato **sales** e un utente chiamato **mary**. Una volta creato il file aggiungetelo alla directory con il seguente comando:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f populate.ldif
```

Vi verrà chiesto di inserire la password che avete definito per l'account di amministratore di LDAP quando avete installato il server. Fate attenzione, perché in questa maniera la password di mary rimane visibile in chiaro all'interno del file: non è una grande idea. In ogni caso la creazione manuale di file LDIF non è ovviamente la maniera più comoda per gestire gli account degli utenti, quindi cerchiamo di trovare qualche strumento di livello superiore. Innanzitutto il pacchetto **ldapscripts** fornisce un certo numero di script di shell che semplificano l'utilizzo dei comandi standard per la gestione di account utente e gruppi. Potete installarli usando il comando:

```
$ sudo apt-get install ldapscripts
```

Dovrete modificare il file di configurazione ***/etc/ldapscripts/ldapscripts.conf*** per assegnare il naming context della vostra directory (**dc=example,dc=com** nel nostro caso) e forse anche qualcos'altro. Le cose più importanti che ho modificato io sono:

```
SUFFIX="dc=example,dc=com"
GSUFFIX="ou=Groups"
USUFFIX="ou=People"
MSUFFIX="ou=Machines"
BINDDN="cn=admin,dc=example,dc=com
```

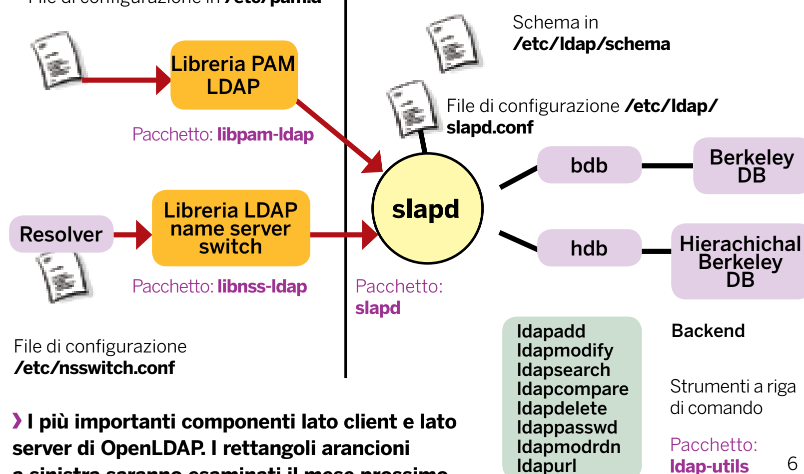
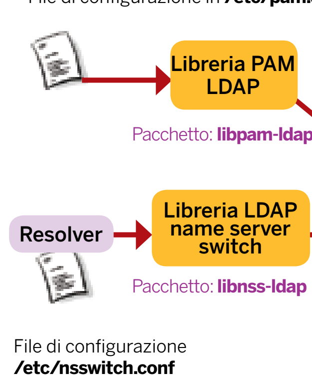
L'ultima riga specifica l'account utente che intendo usare per autenticarmi. Dovrete anche inserire la password amministrativa LDAP nel file **etc/ldapscripts/ldapscripts.passwd** (definito dal parametro **BINDPWDFILE** in **ldapscripts.conf**):

```
sudo echo -n mysecret > /etc/ldapscripts/ldapscripts.passwd
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

Il comando **echo -n** è importante: con un carattere di a capo alla fine il file non funzionerebbe. Possiamo ora aggiungere un utente alla directory in maniera molto più semplice:

```
ldapadduser jane sales
Successfully added user jane to LDAP
Successfully set password for user jane
```

Se non funziona date un'occhiata al file di log ***/var/log/ldapscripts.log*** per avere qualche indicazione. Il pacchetto **ldapscripts** comprende molti altri script utili, tra cui **ldapaddgroup**, **ldapsetpasswd**, **ldapmodifyuser** e così via. Se vi interessa semplicemente una soluzione a riga di comando per gestire gli account degli utenti in LDAP questi script sono quello che cercate.

› I più importanti componenti lato client e lato server di OpenLDAP. I rettangoli arancioni a sinistra saranno esaminati il mese prossimo

Ricerche

Le directory sono state inventate per cercarci le cose. Dalla riga di comando possiamo usare **ldapsearch**. Nel seguente esempio cerchiamo l'UID numerico di **jane**:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane' uidNumber
dn: uid=jane,ou=People,dc=example,dc=com
uidNumber: 10001
```

Abbiamo iniziato la ricerca da **dc=example,dc=com** e stiamo cercando gli elementi che hanno un attributo chiamato **uid** con valore **jane**. Per ogni elemento che corrisponde (nel nostro caso uno solo) stampiamo solo l'attributo **uidNumber**. È possibile stampare più di un attributo:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane' uidNumber loginShell
```

oppure visualizzare tutti gli attributi degli elementi che corrispondono ai criteri di ricerca:

```
$ ldapsearch -x -LLL -b dc=example,dc=com 'uid=jane'
```

Strumenti grafici

Se preferite uno strumento grafico ne troverete molti che vi permettono di esplorare e gestire una directory LDAP. Uno di questi è **LAT** (*LDAP Administration Tool*) che si trova nei repository Ubuntu, quindi l'installazione è molto semplice:

```
$ sudo apt-get install lat
```

Occorre però notare che si tratta di un'applicazione .NET, quindi l'installazione si porterà dietro il runtime se già non è installato sulla macchina. LAT è un eccellente piccolo strumento per la gestione di utenti e gruppi, per esplorare e fare ricerche all'interno della directory e per esaminare schemi. Vi raccomando di dedicargli un po' di tempo. Altri strumenti grafici che vale la pena di esaminare sono **phpLDAPAdmin** (un'applicazione Web) e **Apache Directory Studio**. Qui si conclude la prima parte di questo tutorial. Ora abbiamo una directory LDAP funzionante che possiamo utilizzare come archivio centralizzato per l'autenticazione. Il mese prossimo esaminerò le modifiche lato client necessarie per utilizzare la directory. Ho anche intenzione di dare un'occhiata alla questione della replicazione della directory, per evitare un *single point of failure* e/o per distribuire il carico. LXP

L'angolo di Android



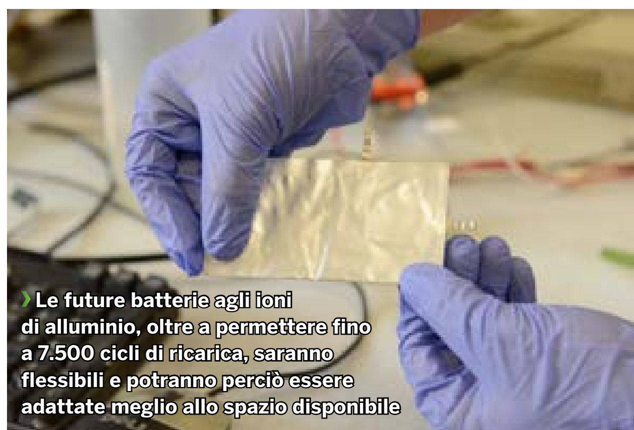
News, recensioni e guide sul sistema operativo libero per smartphone

Se hai news da segnalarci o dei commenti scrivici ad angolo_android@linuxpro.it

Carichi in un minuto

Le nuove batterie promettono di ridurre al minimo i tempi di ricarica di tablet e smartphone

L'autonomia rimane uno dei punti deboli dei moderni dispositivi elettronici. Potremmo anche avere uno smartphone con processore a otto core e 4 GB di RAM, ma se alle sei di sera questo rimane senza autonomia, tanto valeva conservare il vecchio Nokia con cui una decina di anni fa era possibile raggiungere anche una settimana in standby. Non che in questi anni non ci siano stati progressi, l'attuale tecnologia delle batterie agli ioni di litio per esempio ha eliminato l'effetto memoria e gestisce senza problemi fino a 1.000 cicli di carica-scarica completi, fornendo oltretutto il doppio di energia rispetto alle batterie della generazione precedente. Il problema è che in questo stesso periodo gli schermi si sono triplicati come dimensioni e i processori sono diventati almeno dieci volte più potenti. Per questo motivo rappresenta un'ottima notizia l'annuncio dei ricercatori della Stanford University che, sotto la guida del dottor Hongjie Dai, hanno sviluppato un nuovo tipo di batteria agli ioni di alluminio che può essere ricaricata in un minuto. Le celle agli ioni di alluminio hanno anche il vantaggio di essere meno costose rispetto a quelle agli ioni di litio e di essere molto sicure visto che non possono esplodere o prendere fuoco. In realtà la ricerca su questa tecnologia non è nuova, ma fino a oggi non si era riusciti ad arrivare a superare i cento cicli di carica e scarica, troppo pochi per qualsiasi dispositivo elettronico. Grazie però all'utilizzo di un materiale nobile come la grafite usata all'interno del catodo, è ora possibile raggiungere fino a 7.500 cicli completi, quindi oltre sette volte di più rispetto alle attuali batterie agli ioni di litio. Il prototipo sviluppato dal gruppo di lavoro del dottor Hongjie Dai è anche flessibile e questo significa che potrà essere adattato all'interno dei dispositivi elettronici sfruttando ogni spazio e permettendo così di aumentarne la capacità complessiva. Peccato che ci siano ancora un paio di problemi da risolvere: il primo riguarda il voltaggio, visto che gli attuali prototipi sviluppano un massimo di due volt, circa la metà di quelli necessari a tablet e smartphone. Inoltre la densità energetica è ancora sottodimensionata rispetto alle necessità e alle batterie attuali. Si tratta però di problemi che dovrebbero essere risolti in tempi brevi, secondo gli stessi ricercatori non è impossibile pensare a una prima commercializzazione già dal prossimo anno. LXP



› Le future batterie agli ioni di alluminio, oltre a permettere fino a 7.500 cicli di ricarica, saranno flessibili e potranno perciò essere adattate meglio allo spazio disponibile

Android su Linux

App Runtime Control (ARC) è il nome di un software che Google ha realizzato per consentire agli sviluppatori di provare le proprie app prima di inviarle sul Play Store. Si tratta di un vero e proprio emulatore Android che può girare direttamente all'interno del browser Chrome e quindi su tutti i principali sistemi operativi, cioè Linux, Mac OS o Windows. Basta scaricarla direttamente dal Chrome Web Store, avere a disposizione un file APK e scegliere la versione. In realtà non tutti i file APK sono compatibili con Arc, ma noi siamo riusciti a far girare senza problemi giochi anche pesanti come Candy Crush o HillClimb. LXP

Asus Zenfone 2

Un phablet con schermo da 5,5 pollici di ottima qualità, finalmente alla portata di molti

La corsa alle dimensioni sempre maggiori degli schermi non sembra volersi arrestare, ma purtroppo allo stesso tempo anche i prezzi di quelli che gli inglesi definiscono phablet (cioè vie di mezzo tra smartphone e tablet) raramente scendono sotto i 500 €. E non ci riferiamo solo al Galaxy Note di Samsung, ma anche al Nexus 6, il primo smartphone di Google con schermo da 6 pollici ma anche dal prezzo superiore ai 600 €. C'era perciò molto attesa riguardo al phablet di Asus, lo **Zenfone 2** che con il suo schermo da 5,5'' e 170 grammi di peso entra di diritto nella categoria degli ibridi smartphone/tablet. Che Asus badi molto al prezzo lo si capisce dall'offerta: Zenfone 2 è infatti disponibile in tre versioni, con prezzi che variano dai 199 ai 349 €. Noi abbiamo potuto provare il modello intermedio, lo **ZE550ML**, che può già essere acquistato sul sito online di Asus a 249 € e che a noi sembra il migliore come rapporto qualità/prezzo. Per realizzare Zenfone 2, Asus ha sfruttato alcune idee della concorrenza, e secondo noi ha fatto solo bene. Così per esempio i due tasti del volume si trovano sul retro per non ingombrare ulteriormente il davanti e lo smartphone può essere acceso e spento con un doppio tap (e qui l'ispirazione è chiaramente LG che per prima ha inserito il doppio tap e i tasti sul retro). La cover posteriore in plastica presente dà un effetto spazzolatura simile al metallo e può essere staccata per inserire sia la scheda di memoria che le due SIM. Purtroppo la batteria non è rimovibile, ma vista la capacità da ben 3000 mAH non è un grande problema. Il peso di 170 grammi non è tra i più contenuti ma offre una buona sensazione di solidità e comunque più che sul peso è forse meglio soffermarsi sulla lunghezza complessiva del dispositivo superiore ai 15 cm. Lo schermo IPS da 5,5'' ha una risoluzione HD, cioè da 1280x720 pixel, il rivestimento è l'ottimo Gorilla Glass di terza generazione. I colori dello schermo sono naturali e grazie all'app **Splendid** di Asus è possibile personalizzarne la temperature. La luminosità massima è decisamente elevata, così come quella minima e questo può essere un problema per chi è abituato a leggere al buio. La regolazione automatica della luminosità non è perfetta, ma questo è un problema che affligge da sempre la maggior parte degli smartphone Android. Sinceramente avevamo qualche riserva su una risoluzione così "limitata" soprattutto considerando che sui più recenti modelli da 5,5" si va ben oltre il Full HD. In realtà siamo rimasti piacevolmente soddisfatti dalla resa del display durante l'utilizzo quotidiano, anche allargando immagini in alta definizione non abbiamo infatti notato particolari aberrazioni. Chi comunque volesse a tutti i costi lo schermo Full HD, insieme a un processore più veloce, 4 GB di RAM il doppio della memoria e la connessione NFC, potrà acquistare la versione superiore **ZE550ML** a 349 €.



Processore Intel

Per riuscire a risparmiare sui costi Asus ha scelto di dotare Zenfone 2 di uno dei nuovi processori Intel Atom Z3560 quad core da 1,8 GHz, con supporto ai 64 bit e 2 GB di RAM. Il SoC non è alla pari dei più veloci processori di Qualcomm, ma è comunque valido per un prodotto che vuole contenere i costi senza troppe rinunce. Per avere un'idea più precisa delle prestazioni abbiamo eseguito il benchmark Antutu ottenendo un risultato di 42.000 punti, paragonabile per intenderci al Galaxy S5 uscito lo scorso anno. La memoria da 16 GB (11 GB reali) può essere ampliata con altri 64 GB via scheda microSD, per quanto riguarda la doppia SIM, la buona notizia è che la tecnologia è la Dual Sim Dual Active, cioè entrambe le linee telefoniche sono attive contemporaneamente. Quella meno buona è che la Sim secondaria è solo 2G e non può essere usata per la connessione dati. La fotocamera posteriore da 12,6 megapixel è assistita da due illuminatori Led potenti ed efficaci. Le foto sono di buona qualità anche se, come spesso accade con gli smartphone, soffrono le situazioni di bassa luminosità. In questi casi però si può attivare la funzione PixelMaster di Asus che riduce la risoluzione delle foto per interpolare i pixel e realizzare così foto con poco rumore anche in caso di poca luce. L'autonomia non è eccezionale, viste anche le dimensioni dello schermo, ma consente comunque di concludere senza problemi anche una lunga giornata da 18 ore. LXP



› **Le grandi dimensioni complessive richiedono l'utilizzo di due mani per raggiungere tutte le parti del display**

LINUX PRO Giudizio

Asus Zenfone 2

Produttore: Asus
Web: http://www.asus.it
Prezzo: 249 €

Caratteristiche	9
Autonomia	7.5
Prestazioni	7.5
Qualità/prezzo	10

» *Attualmente il miglior rapporto qualità-prezzo disponibile per un phablet.*

Il voto di Linux Pro 8.5



COD. LP141 N 141 - MAGGIO 2014



COD. LP142 N 142 - GIUGNO 2014



COD. LP143 N 143 - LUGLIO 2014



COD. LP144 N 144 - AGOSTO 2014



COD. LP145 N 145 - SETTEMBRE 2014



COD. LP146 N 146 - OTTOBRE 2014



COD. LP147 N 147 - NOVEMBRE 2014



COD. LP148 N 148 - DICEMBRE 2014



COD. LP149 N 149 - GENNAIO 2015



COD. LP150 N 150 - FEBBRAIO 2015



COD. LP151 N 151 - MARZO 2015



COD. LP152 N 152 - APRILE 2015

Se leggi Linux Pro, ti possono piacere anche...

OFFERTA SPECIALE
La guida completa ai droni: come sceglierli e come guadagnarci
COD. IMCS 2
€ 9,90

DRONI
LA PRIMA GUIDA COMPLETA

COD. RGV6
€ 9,90



COD. MICR1
€ 9,90



Completa la tua collezione ordinando gli arretrati a soli € 5,90 cad.
su www.linuxpro.it/arretrati
oppure utilizzando il modulo qui sotto

SCEGLI L'ARRETRATO CHE VUOI ORDINARE

SE VUOI ORDINARE VIA POSTA O VIA FAX, COMPILA QUESTO COUPON

Ritaglia o fotocopia il coupon, invialo in busta chiusa a: **Sprea S.p.A.** Via Torino, 51 20063 Cernusco s/n (MI), insieme a una copia della ricevuta di versamento o a un assegno. Oppure via fax al numero 02.700537672. Per ordinare in tempo reale i manuali collegati al nostro sito **www.linuxpro.it/arretrati.** Per ulteriori informazioni puoi scrivere a **arretrati@linuxpro.it** oppure telefonare allo 02/87158224 tutti i giorni dalle 14.00 alle 18.00

INSERISCI I CODICI delle pubblicazioni che desideri ricevere:

	€
	€
	€
	€
Totale Ordine	€

SCELGO IL SEGUENTE METODO DI SPEDIZIONE:
Indica con una ✗ la forma di spedizione desiderata

☐	Per **una rivista** spedizione tramite **posta tradizionale** al costo aggiuntivo di	**€ 3,90**
☐	Per **due o più riviste** spedizione tramite **Corriere Espresso** al costo aggiuntivo di	**€ 7,00**
	TOTALE COMPLESSIVO	€

Data |_|_|_|_|_|_|　　Firma del titolare________________________

9 771722 616763 50153

NOME |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
COGNOME |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
VIA |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
N° |_|_|_|_|_|　**C.A.P.** |_|_|_|_|_|　**PROV.** |_|_|
CITTÀ |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
TEL. |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
E-MAIL ---------------------------------------

SCELGO IL SEGUENTE METODO DI PAGAMENTO (Indica con una ✗ quello prescelto)

☐ Versamento su **CCP 99075871** intestato a **Sprea S.P.A. ABBONAMENTI Via Torino 51 20063 Cernusco Sul Naviglio MI** (Allegare ricevuta nella busta o al fax)

☐ Bonifico intestato a **Sprea S.P.A. Abbonamenti** sul conto **IBAN IT05 F076 0101 6000 0009 9075 871**

☐ **Carta di Credito** N. |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
(Per favore riportare il numero della Carta indicandone tutte le cifre)
Scad. |_|_|_|_|　CVV |_|_|_|_|_|

|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|
Nome e Cognome del Titolare della carta (può essere diverso dall'abbonato)

☐ CartaSì ☐ postepay ☐ MasterCard ☐ VISA ☐ Diners Club International ☐ American Express

Sony Xperia M2 Aqua

Sony cerca di fare la differenza con uno smartphone impermeabile

La società nipponica ha scelto di mettere sul piatto tutta la sua esperienza acquisita nel settore della telefonia mobile impermeabile, perché un telefono di qualità non può esalare l'ultimo respiro quando finisce nella toilette o nella vasca da bagno.
Il risultato è l'**M2 Aqua**, che unisce il look elegante della serie M2 con la robustezza del modello ZR. A prima vista l'M2 Aqua è molto bello. Ha degli spigoli accattivanti, e le porte per schede microSD, cuffie e caricabatteria sono protette dall'acqua con uno sportello impermeabilizzato con finiture lisce.
Per l'impermeabilizzazione, lo smartphone ha passato i nostri test, conservando la piena usabilità anche quando era bagnato. Il processore Snapdragon 400 quad-core è in grado di gestire più app contemporaneamente in esecuzione senza evidenti rallentamenti. Una delle sue caratteristiche principali è la gamma di opzioni per la fotocamera. Quella posteriore da otto megapixel fornisce un'immagine nitida, supportata da molte tecnologie. Ma il divertimento inizia davvero usando le app. La nostra preferita è **AR photos**. Puntando il telefono verso una superficie uniforme permette di aggiungere alla scena dei divertenti personaggi, dai dinosauri ai folletti. Le immagini sono mozzafiato e possono dare a una foto un tocco di divertente follia. È un peccato che la fotocamera anteriore sia solo da 1,1 megapixel, con conseguente granulosità dei selfie, ma nel complesso è meno terribile di quanto ci saremmo aspettati.
Il problema con l'M2 Aqua è trovare lo spazio necessario per archiviare le foto.
In teoria offre un già scarso spazio di 8 GB ma, tra sistema operativo e app preinstallate, ne rimangono effettivamente disponibili solo 4 GB. Fortunatamente è possibile ovviare al problema grazie allo slot per microSD che supporta schede sino a 32 GB. Tra le app precaricate, quella che c'è piaciuta di più è **Sony Select**, che propone una serie d'interessanti giochi e altre app.
C'è piaciuta anche l'app **News & Weather**, che riassume notizie e meteo della settimana in modo semplice e efficiente. Lo schermo da 4,8 pollici è comodo da leggere e gestire.
L'M2 Aqua può essere riassunto come un solido tuttofare. È bello, ma è più facile apprezzare la sua resistenza all'acqua. Ha delle valide qualità e, pur non eccellendo in qualche specifico settore, è interessante la gamma di funzioni per la fotocamera. Forse la scarsa risoluzione della fotocamera frontale è il vero limite. Offre un buon mix di app Android e di app proprietarie Sony e, probabilmente, non resta che caricare sul telefono i propri giochi preferiti. Non è perfetto, ma mostra elementi validi sufficienti per renderlo un buon acquisto nella fascia media degli smartphone. LXP



Scheda tecnica

» **Sistema operativo** Android KitKat 4.4.2
» **Processore** Snapdragon 400 quad 1,2 GHz
» **Dimensioni** 140 x 72 x 8,6 mm
» **Memoria** 1+8 GB di RAM
» **Peso** 149 grammi
» **Schermo** 4,8''
» **Risoluzione schermo** 540 x 960 pixel
» **Espansione** MicroSD

LINUX PRO Giudizio

Sony Xperia
Produttore: Sony
Web: www.sonymobile.it
Prezzo: 249 €

Caratteristiche	7
Autonomia	8
Prestazioni	8
Qualità/prezzo	8

» *Prezzo ragionevole, impermeabile, ma niente di straordinario in generale.*

Il voto di Linux Pro 7.5

Tornado 348

Kazam, società inglese da poco arrivata in Italia, ha realizzato lo smartphone più sottile al mondo, con un buon rapporto qualità/prezzo e alcune chicche

Kazam è un marchio da poco arrivato in Italia con una gamma di prodotti tra cui spicca il **Tornado 348**, "lo smartphone più sottile al mondo", una sottiletta che ci ha sorpreso per design, prestazioni e autonomia. Appena lo si ha tra le mani sembra quasi si tratti di uno di quei modelli finti da esposizione: non solo è sottilissimo, ma pesa meno di un etto e una volta acceso invece è fantastico!
La differenza con la concorrenza in fondo è di pochi mm, ma vi assicuriamo che il risultato è davvero notevole. L'ottimo schermo Amoled HD da 4,8" e la parte posteriore sono protetti con Gorilla Glass 3, che ha una elevata resistenza agli urti e ai graffi e, con la sua superficie lucida, rende lo smartphone estremamente elegante.
È così sottile che inizialmente bisogna fare attenzione per non farselo sfuggire di mano, ma la scocca è in metallo e Kazam ti garantisce lo schermo per un anno contro le rotture accidentali.
Il Tornado 348 monta un processore octa-core Mediateck da 1,7 GHz, 1 GB di RAM e una memoria flash da 16 GB, di cui circa una decina liberi per l'utente. Non c'è spazio per una scheda di memoria microSD aggiuntiva, ma il supporto OTG permette la connessione di chiavette USB e altri accessori tramite la porta micro USB che usi per la ricarica. L'interfaccia è quella classica di Android 4.4 che gira in maniera assolutamente fluida, il processore a otto core non mostra alcun tentennamento neanche con i video in alta risoluzione e permette di giocare anche durante un pesante download. Kazam ha aggiunto solo qualche applicazione come la torcia, un file manager, un'app per gestire la radio FM, una per un veloce editing video e quella per l'assistenza da remoto. Supporta alcune interessanti gesture come la riattivazione con un doppio tocco sul display (in stile LG) o quelle che consentono di avviare specifiche applicazioni direttamente dalla modalità inattiva disegnando forme geometriche sullo schermo.
La fotocamera posteriore da 8 Mpixel ha il pregio di non sporgere, mantenendo pulite le linee dello smartphone, consente buoni scatti e riprese video in Full HD.
È possibile regolare la zona di messa a fuoco e quella per la regolazione dell'esposizione automatica per bilanciamento ottimale della luce sulle foto. La fotocamera frontale è da ben 5 Mpixel e permette di scattare selfie inquadrandoti e mettendo le mani a "V", oppure sorridendo. Purtroppo non è privo di difetti: la mancanza della connessione 4G si fa sentire durante la navigazione tramite rete mobile, così come la connessione NFC per l'associazione ad accessori e i micro pagamenti. Il Tornado è realizzato con cura, ha ottime prestazioni generali, un buon rapporto qualità/prezzo e come biglietto da visita ci ha colpito in positivo. LXP



› Essendo così sottile c'era il rischio che l'autonomia fosse il suo punto debole, invece la batteria da 2.050 mAh sorprende per la durata. La fotocamera da 8 Megapixel sul retro si integra perfettamente con l'eccellente design del dispositivo

Scheda tecnica

- **» Sistema operativo** Android 4.4.2 KitKat
- **» Processore** MediaTek6592 a 1,7 GHz
- **» Dimensioni** 140 x 68 x 5,1 mm
- **» Memoria** 1+16 GB di RAM
- **» Peso** 96 grammi
- **» Schermo** 4,8''
- **» Risoluzione schermo** 1280 x 720 pixel
- **» Espansione** No

LINUX PRO Giudizio

Kazam Tornado 348

Produttore: Kazam
Web: www.kazam.mobi/it-it
Prezzo: 299 €

Caratteristiche	7
Autonomia	9
Prestazioni	8
Qualità/prezzo	8

» Bello e leggero ma solo se non vi interessa la connessione 4G.

Il voto di Linux Pro 8

Jolla Sailfish OS

Dalla lontana Finlandia arriva un nuovo smartphone ma soprattutto un nuovo sistema operativo...

Il primo smartphone **Jolla** con sistema operativo **Sailfish OS** ha visto la luce nel dicembre del 2013 e finalmente arriva in distribuzione anche in Italia. Scopriamolo partendo dal design che è piuttosto originale. Decisamente squadrato, in controtendenza con gli angoli smussati di molti concorrenti, il Jolla si presenta come se fosse costituito da due pannelli appaiati. Il frontale propone uno schermo in Gorilla 2 Glass praticamente privo di cornice, mentre la cover posteriore intercambiabile presenta angoli secchi sui lati lunghi e una pronunciata bombatura sui lati corti.
Le dimensioni contenute e il peso decisamente ridotto fanno sì che si impugni perfettamente permettendo di controllarlo agevolmente con una sola mano.
Per quanto riguarda le caratteristiche hardware generali, siamo in una fascia intermedia. Monta un processore Qualcomm Dual Core a 1,4 GHz supportato da 1 GB di RAM; una buona dotazione che gli permette di essere veloce e fluido nell'utilizzo quotidiano, ma non si pone certo al vertice della categoria. La memoria interna invece è generosa in quanto offre 16 GB di spazio di archiviazione espandibile tramite lo slot per MicroSD. La fotocamera frontale è da 2 Mp, e svolge bene la sua funzione per gli ormai immancabili selfie e per le video chiamate, mentre quella posteriore è da 8 Megapixel. La qualità delle foto non ci ha né entusiasmato, né delusi. Sia in condizioni di buona luce sia con luce scarsa abbiamo registrato scatti nella media, ma dobbiamo ammettere che la stessa qualità l'abbiamo riscontrata anche con smartphone decisamente più a buon mercato. Ci sono anche alcune mancanze, per la verità legate più al software che all'hardware, come per esempio l'impossibilità di impostare la risoluzione della foto, ma ci sono anche delle caratteristiche interessanti e nel complesso il comparto fotografico si presenta all'altezza. Ma ciò che distingue veramente il Jolla è il suo sistema operativo Sailfish OS. Mettiamo per un attimo da parte i dubbi sul fatto che nel mercato ci sia o meno spazio per un nuovo sistema operativo da affiancare ai colossi Android e iOS e vediamo di esaminarlo. La caratteristica principale di Sailfish è quella di non avere tasti fisici o a schermo ma tutta la navigazione all'interno dei vari menu avviene tramite gesti, strisciando dall'alto, dal basso, a destra e a sinistra. Non è difficile e una volta presa la mano è addirittura ancora più semplice usarlo con una sola mano rispetto a un OS tradizionale, ma il periodo d'apprendimento può essere lungo, soprattutto se si possiede uno smartphone da tanto tempo. Dedichiamo infine una parola allo Store, punto dolente di tutti gli OS alternativi (e non solo, vero Microsoft?). Le app non sono molte e soprattutto la navigazione e la ricerca all'interno dello store non appare semplice. Jolla ha aggiunto la possibilità d'installare gli apk delle app di Android. Questo amplia decisamente la disponibilità di applicazioni ma attenzione: non pensate di avere a disposizione tutte le app del Play Store di Google, solo alcune sono realmente compatibili. LXP



› **La fotocamera posteriore da 8 Megapixel garantisce scatti di buona qualità. L'applicazione presenta lacune ma dispone di caratteristiche come la scelta della sensibilità alla luce o il ritardo di scatto**

Scheda tecnica

» **Sistema operativo** Sailfish OS 1.1.1.27
» **Processore** Qualcomm Dual a 1,4 GHz
» **Dimensioni** 31 x 68 x 9,9 mm
» **Memoria** 1+16 GB di RAM
» **Peso** 141 grammi
» **Schermo** 4,5"
» **Risoluzione schermo** 960×540 pixel
» **Espansione** microSD

LINUX PRO Giudizio

Jolla Sailfish OS

Produttore: Jolla
Web: www.jolla.com
Prezzo: 299 €

Caratteristiche	8
Autonomia	8
Prestazioni	8
Qualità/prezzo	8

» *Un interessante esperimento che però non offre troppo di nuovo.*

Il voto di Linux Pro **8**

Recensioni

Tutte le novità in campo software e hardware testate e valutate ogni mese dai nostri laboratori

Se vuoi segnalarci qualche novità scrivi a recensioni@linuxpro.it

Una breve legenda

Ogni test di questa sezione è accompagnato da un giudizio che riassume con quattro indici numerici le principali qualità dell'applicazione o del prodotto hardware messo alla prova. I laboratori di Linux Pro assegnano un voto da 1 a 10 alle seguenti categorie:

Caratteristiche: fornisce tutte le funzioni di cui abbiamo bisogno? È innovativo?
Prestazioni: esegue in maniera efficiente le sue funzioni? È veloce e affidabile?
Facilità d'uso: dispone di un'interfaccia grafica chiara e facilmente fruibile? La documentazione che lo accompagna è sufficientemente completa ed esaustiva?
Qualità/prezzo: ha un prezzo competitivo? Vale i soldi richiesti per il suo acquisto?

Il nostro giudizio viene poi riassunto da un voto finale, espresso anche graficamente. Ecco la legenda dei voti:

10 Nulla da eccepire. Un prodotto praticamente perfetto.
8-9 Un buon prodotto. I pochi difetti presenti non sono gravi.
6-7 Compie il suo lavoro ma necessita di ulteriori sviluppi.
5-4 Deve migliorare prima di raggiungere un voto sufficiente.
1-3 Un completo disastro. Gli sviluppatori devono tornare alla fase di progettazione.

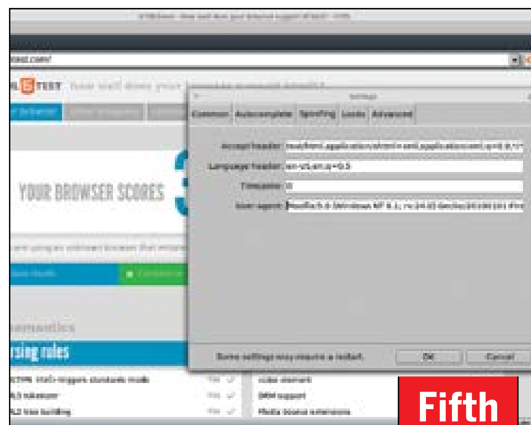
Ricordiamo infine che i software citati nelle sezioni Confronto e Da non perdere sono spesso presenti nel DVD sotto la voce "Rivista" sotto forma di codice sorgente o binario.

QUESTO MESE...


Chromebook 2



Ruby



Fifth

Toshiba Chromebook 2

Finalmente uno spettacolare display Full HD da 1080p su un Chromebook che spicca per qualità costruttiva e design

In breve

» Un ottimo PC che si distingue per il design e il display Full HD. La batteria non ha però una durata eccezionale.

Specifiche

» CPU: 2,16 GHz Intel Celeron N2840
» Grafica: Intel HD
» RAM: 4 GB
» Schermo: 13.3", 1.920x1.080
» Spazio: 16 GB SSD
» Porte: 1x USB 3.0, 1x USB 2.0, HDMI, scheda SD
» Comunicazione: Wireless-AC, Bluetooth 4.0, HD Webcam
» Dim: 320x213x 19,3 mm, 1,34 Kg

Sulla carta, il **Toshiba Chromebook 2** è migliore rispetto al modello precedente su tutti i fronti. Oltre a essere più piccolo e più sottile, continua a supportare un display da ben 13,3 pollici. Grazie al nuovo design privo di ventole, Toshiba è riuscita a ridurre drasticamente le dimensioni di 20,3x19,3 millimetri, facendo scendere il peso da 1,5 Kg a 1,3 Kg. Al di là dei numeri, però, questo Chromebook impressiona fin dal primo sguardo. Il bordo sottile spicca su tutto, tanto da farlo sembrare uno dei portatili con spessore tra i più ridotti in circolazione. Il design è stato leggermente modificato, operando una passata di vernice opaca e squadrando gli angoli. L'unico problema di progettazione che salta subito agli occhi è la cover posteriore bianca. Messa a confronto con il resto, stona fortemente, facendo decadere l'eleganza generale del prodotto. La tastiera è molto comoda: i tasti sono ben ammortizzati e non fanno dolere le dita neppure dopo ore di battitura. Sotto di essa troviamo un trackpad preciso e reattivo. Toshiba ha dotato questo PC di un processore Intel Celeron N2840 da 2,16 GHz, 4 GB di RAM e un display da 13.3" a 1.920x1.080 pixel. A dire la verità, quando abbiamo ricevuto in prova questo portatile, l'esperienza con il predecessore ci ha fatto partire con il piede sbagliato. Infatti, considerando che il processore è lo stesso del Chromebook 2 di Samsung, abbiamo pensato che le prestazioni fossero deludenti come in passato. Tuttavia è bastato accenderlo per ricrederci del tutto. Infatti, il portatile è riuscito ad aprire un flusso di ben 18 schede nel browser senza accennare al minimo rallentamento. In più, lo abbiamo sottoposto a un ciclo continuo di streaming a 1080p a cui ha reagito senza la minima incertezza.

Performance di tutto rispetto

Gli unici problemi relativi alle prestazioni li abbiamo riscontrati durante una partita a Bastion (un browser game) a piena risoluzione. Nel momento in cui ci siamo avventurati in ambienti più particolareggiati, il frame rate è sceso a 10 fotogrammi al secondo. In realtà, non siamo rimasti poi così tanto stupiti. Un comportamento del genere è del tutto normale, se consideriamo che la scheda video del Chromebook è integrata, mentre il nostro gioco ne vorrebbe una dedicata. Passando invece al profilo più strettamente tecnico, se c'è una cosa su cui i Chromebook sono stati fino a ora deficitari, è proprio la risoluzione dello schermo. Per troppo tempo questi PC sono stati equipaggiati con display che non riuscivano a superare i 1.366x768 pixel. Il Toshiba Chromebook 2, sotto questo aspetto, rompe gli schemi, offrendo oltre a una risoluzione in Full HD, uno schermo di ottima fattura e qualità. Invece di utilizzare un pannello TN come la maggior parte dei PC simili, Toshiba ha sfruttato un



› Il nuovo Chromebook di Toshiba beneficia del design senza ventole

IPS che rende i colori più vivaci e produce neri profondi. Il risultato è un display perfetto per vedere film in Full HD. Naturalmente c'è anche il rovescio della medaglia che può essere identificato con la minor durata della batteria. Infatti, il Chromebook 2 ha un'autonomia di sole 6 ore e 26 minuti con un utilizzo di sola navigazione Web e qualche videochiamata. Di per sé non è un risultato particolarmente negativo, ma i computer di questo genere ci hanno abituati a ben altre tempistiche. Basta pensare all'Acer C720 che ha una durata della batteria pari a 8 ore e 30 minuti o al Samsung Chromebook 2 che si spegne dopo 7 ore e 32 minuti. LXP

Caratteristiche

Display IPS
Il Toshiba, grazie al display Full HD, rompe gli schemi con gli altri Chromebook.

Trackpad e tastiera
Una tastiera comoda e un trackpad preciso sono gli strumenti che utilizziamo per sfruttare questo PC.

LINUX PRO Giudizio

Toshiba Chromebook 2

Sviluppatore: Toshiba
Web: www.toshiba.com
Prezzo: 340 € circa

Caratteristiche	6/10
Prestazioni	8/10
Facilità d'uso	9/10
Qualità/prezzo	8/10

» *Pur avendo un'autonomia piuttosto scarsa, può contare su uno splendido display Full HD da 13".*

Il voto di Linux Pro 8

Civ: Beyond Earth

Gli appassionati della saga di Sid Meier non rimarranno delusi. L'ultimo capitolo merita senza alcun dubbio di essere giocato

In breve

» Un gioco strategico che non manca di appassionare. Beyond Earth, per certi versi, è molto simile a Civ 5, ma ha in sé un nucleo di novità che lo rende davvero divertente in ogni circostanza.

Specifiche

» OS: Steam OS, Ubuntu 14.04

» CPU: Intel Core i3 o AMD A10

» Memoria: 4 GB

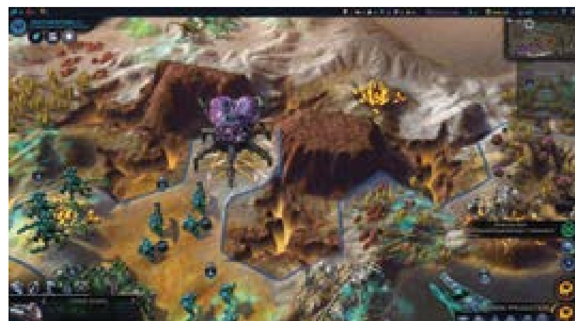
» Gfx: Nvidia Geforce 260

» HDD: 8 GB

Civilization: Beyond Earth inizia come un vero e proprio capitolo della saga. Tuttavia, se avete perso i titoli precedenti, non dovete preoccuparvi. Pur ricalcando la filosofia portante della serie, infatti, si tratta di qualcosa di completamente diverso. Al di là delle similitudini, troverete molte novità che prendono in considerazione le unità strategiche, gli alberi tecnologici, i leader, le civiltà e una serie di altre cose nascoste sotto la superficie del gioco. Quando iniziamo, colonizzando un nuovo spazio che sulla mappa viene mostrato completamente nero, ci sentiamo proprio come se stessimo giocando a Civ 5. Detto questo, se siete in cerca di un sequel diretto o di un aggiornamento in chiave moderna del primo titolo della saga lanciata nel 1999, resterete delusi. Rispetto alle classiche civiltà terrestri, in questo gioco troviamo una grande quantità di alieni, alcuni più aggressivi di altri. Un altro aspetto che riveste un ruolo di novità è il livello orbitale. Infatti, è possibile costruire e lanciare satelliti che vi permetteranno di avere diversi vantaggi nella strategia generale del gioco. La cattiva notizia, però, è che divertendosi con questo titolo, è facile concepirlo come un semplice aggiornamento di Civ 5. Gli amanti di quest'ultimo capitolo, infatti, si sentiranno a casa propria, prendendo subito confidenza con gli strumenti a disposizione. Al di là delle somiglianze, comunque, Beyond Earth riveste un ruolo di innovatore nella saga, dovuto soprattutto all'ambientazione futuristica che gli sviluppatori sono riusciti a creare in tutto il suo splendore.



› ***Beyond Earth*** **permette di giocare con cinque modalità, ognuna delle quali basa la vittoria su diversi tipi di strategie**



› **La struttura del gameplay è molto simile a quella dei giochi passati, ma l'ambientazione rende *Beyond Earth* davvero unico**

Qualcosa di più rispetto a Civ 5

In base alla civiltà che decidete di impersonare, avrete più o meno vantaggi durante tutto l'arco del gioco. I pro e i contro, di solito, si fondono in un mix ben congeniato che permette di divertirsi indipendentemente dalla fazione di cui si portano i colori. In alcune occasioni abbiamo però notato un certo dislivello tra le varie civiltà. Con alcune, il gioco è decisamente più difficile, tanto da far fatica a tenere testa ai nemici sempre sull'orlo della vittoria. Con altre, invece, il gameplay è più bilanciato e se per certi versi le sfide possono sembrare meno accattivanti, si ha uno sviluppo molto più lineare e tendenzialmente credibile. Beyond Heart offre cinque condizioni di vittoria, due delle quali sono piuttosto simili tra loro. *Contact* pone come obiettivo lo scoprire un segnale alieno che sbloccherà il segreto della vostra specie, facendovi conoscere un'antica razza. *Domination* è di facile interpretazione: bisogna conquistare tutte le civiltà antagoniste, schiacciando le rispettive capitali. *Priomised Land* ed *Emancipation*, invece, sono due facce della stessa medaglia. È necessario puntare sulla ricerca tecnologica per arrivare alla salvezza o al dominio del mondo. Infine abbiamo la modalità *Harmony*, dedicata a chi vuole giocare senza sparare un colpo. Consiste nel ricercare tecnologia aliena per creare una condizione che unirà la vostra coscienza con quella del pianeta. In definitiva, Beyon Heart è uno dei giochi della saga meglio riusciti, che tuttavia potrà appassionare solo gli amanti del genere strategico. Infatti, si riferisce a una nicchia ben strutturata di giocatori che difficilmente riuscirà ad aprirsi ad altre fasce di player. LXP

LINUX PRO Giudizio

Civilisation: Beyond Earth

Sviluppatore: Aspyr
Web: GameAgent.com
Prezzo: 39,90 €

Giocabilità	9/10
Grafica	8/10
Longevità	8/10
Qualità/prezzo	8/10

» *Anche se la somiglianza con Civ 5 si sente, Beyond Earth è un gioco ricco di sorprese.*

Il voto di Linux Pro **8**

ThinkPad Yoga 11e

Un notebook che fa della resistenza il suo cavallo di battaglia. Pensato per sopportare urti, pressione e polvere, ha solo un difetto: l'autonomia

Specifiche

- » CPU: 1,83 GHz Intel Celeron N2930 (quad-core)
- » GPU: Intel HD
- » RAM: 4 GB DDR3 (1.333 MHz)
- » Schermo: 11.6", 1.366 x 768 HD LED
- » Disco: 16 GB SSD (eMMC)
- » Porte: HDMI 1.4, USB 3.0, USB 2.0, lettore SD
- » Comunicazione: Intel 7260 802.11ac, Bluetooth 4.0
- » Video: Webcam 720p HD
- » Peso: 1,5 Kg
- » Dimensioni: 300x216x22 mm

Lo **Yoga 11e** è uno dei Chromebook più robusti in circolazione. Progettato per l'uso in ambienti dove la solidità è il primo aspetto a cui guardare, vanta bordi gommati e cerniere extra resistenti. Tanta forza, però, fa lievitare il prezzo ben oltre il limite imposto dai principali concorrenti. Considerando questo aspetto, quindi, c'è da chiedersi se la qualità costruttiva sia in linea con il costo d'acquisto. Il design continua a essere minimalista, riprendendo la filosofia già vista in altri ThinkPad. La plastica del telaio è in nero opaco e, a parte il logo in argento, non ci sono altre varianti sul tema. Le dimensioni sono di 300x216x22 millimetri, mentre il peso è di 1,5 Kg. Dati, questi, che rendono lo Yoga un po' più pesante del Lenovo N20p. Come altri notebook della serie Yoga, il display di questo modello può ruotare di 360 gradi, così da permettere una maggiore versatilità. Il sistema basculante, inoltre, consente di utilizzarlo in ben quattro modalità: laptop, tablet (ripiegandolo completamente all'indietro), stand (aperto a 180°) e tenda. Quest'ultima consiste nel disporre il Chromebook con un'angolatura di 270 gradi, in piedi sul profilo del display e della tastiera. Sebbene lo Yoga 11e sia piuttosto comodo da usare in versione tablet, con questa modalità risulta un po' limitato. Infatti, l'interfaccia di Chrome OS dispone di densi ammassi di piccoli pulsanti, in particolare sulla destra del browser, che risultano molto difficili da usare con i comandi touch. Diversamente da altri tablet, lo Yoga 11e è stato costruito per resistere alla frenesia della vita quotidiana e agli ambienti dove i portatili simili non potrebbero nemmeno avvicinarsi. Infatti, è in grado di sopportare forti pressioni, umidità, vibrazioni, basse temperature e polveri di qualsiasi natura. Venendo alle prestazioni, possiamo dire che le funzionalità multitasking si scontrano con l'hardware piuttosto limitato. Utilizzando Google Docs, insieme allo streaming di musica con un'altra dozzina di schede aperte, il computer inizia a risentire di qualche rallentamento sia nella gestione delle attività, sia nella ricezione dei comandi.



Ottimo display

A bilanciare le prestazioni tutt'altro che brillanti, ci pensa però il display. Siamo davanti a un ottimo 11,6 pollici con risoluzione di 1366x768 pixel, particolarmente contrastato e nitido. I colori risaltano durante la visione dei video in HD e il testo compare sempre chiaramente leggibile. L'unico difetto riguarda gli angoli di visione. Se ci spostiamo leggermente di lato, i colori e la qualità iniziano a perdere di consistenza. Il touchscreen risponde prontamente ai comandi, anche se nel nostro caso abbiamo riscontrato un problema con il pinch-to-zoom. Gli altoparlanti sono potenti e consentono di ascoltare l'audio dei video e la musica con buona soddisfazione. La tastiera è costruita per resistere a tutto, ma non per questo è scomoda da utilizzare. I tasti sono ammortizzati e hanno un fine corsa ben tarato. Il touchpad è ampio e reattivo, lo scorrimento con due dita è immediato e basta sfiorarlo per impartire i vari comandi. La durata della batteria è deludente. Lo Yoga 11e è il computer con autonomia più bassa tra i Chromebook provati. Con la luminosità al 30% e una dozzina di schede aperte, questo notebook si spegne dopo appena 6 ore e 21 minuti. A livello di connettività possiamo contare su una HDMI 1.4, una USB 3.0, USB 2.0 e un lettore di schede SD 4 in 1. Nella parte frontale, troviamo poi una Webcam 720p HD. Volendo tirare le fila, questo modello non è indenne da qualche difetto, ma dalla sua parte c'è sicuramente la solidità e l'ottima qualità costruttiva. Il prezzo non è alla portata di tutti, ma considerando le doti di cui dispone, possiamo dire che il rapporto tra qualità e costi c'è tutto. LXP



› **Lo Yoga può vantare davvero diverse modalità d'uso. Basta semplicemente ruotare il display e il gioco è fatto**

LINUX PRO Giudizio

Lenovo ThinkPad Yoga 11e

Sviluppatore: Lenovo
Web: www.lenovo.com
Prezzo: 489 € circa

Caratteristiche	6/10
Prestazioni	7/10
Facilità d'uso	9/10
Qualità/prezzo	6/10

» *Un design solido e robusto che offre anche una tastiera comoda da usare. Peccato solo per l'autonomia.*

Il voto di Linux Pro **7**

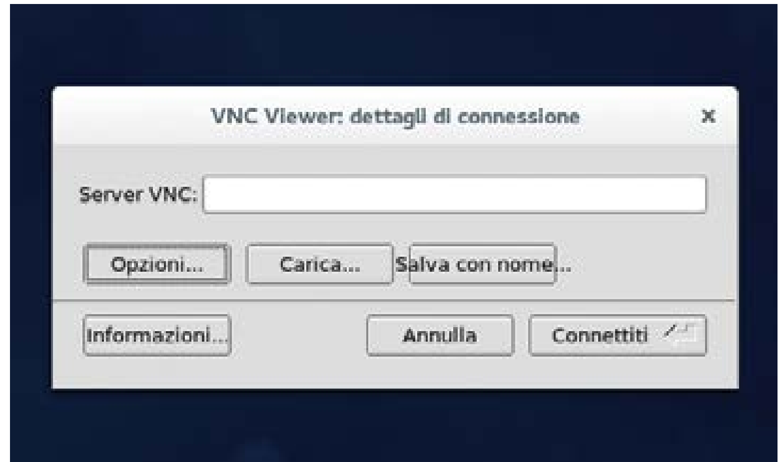
TigerVNC 1.4.0

Linux Pro ha provato l'ultima versione di uno dei VNC più celebrati, che nonostante tutto presenta ancora svariati problemi di prestazioni

In breve

» Un client per desktop remoti che utilizza il protocollo VNC. Guardate anche Vino e Krfb/krdc.

Nel 2009, con l'aiuto di Red Hat, il server **TigerVNC** nacque come fork di TightVNC. L'obiettivo del progetto era rendere possibile l'esecuzione di applicazioni grafiche 3D in modalità a schermo intero su desktop remoto. Poco dopo il suo debutto, TigerVNC è stato adottato dalla distro Fedora con l'implementazione di VNC. Nel corso degli anni, il progetto ha però perso interesse e alcuni sviluppatori hanno abbandonato la nave. I rilasci di TigerVNC, pertanto, sono diventati cosa rara ed ecco perché di recente, avendo annunciato una major release, siamo stati attirati dall'idea di provarla. Alcune caratteristiche, come la compressione e l'accelerazione JPEG, sono stati presi in prestito da altre soluzioni. Il progetto in questione, peraltro, funziona anche su Windows e Mac OS X. TigerVNC include un visualizzatore cross-platform barebone che non integra alcuna caratteristica particolare già vista in altri client, come Remmina o Vinagre. Non c'è alcuna lista di server VNC presenti in rete, né la possibilità di sfruttare i segnalibri. Quello che viene offerto, invece, è una semplice finestra con una casella di testo in cui inserire l'indirizzo IP del server a cui volete connettervi. Questa ristrettezza nell'interfaccia grafica, tuttavia, non significa che TigerVNC sia deficitario dal punto di vista delle impostazioni. Facendo click sul tasto **Opzioni**, infatti, avrete la possibilità di mettere mano a parecchie funzioni che consentono di gestire la connessione.
Per configurazione predefinita, il software è impostato per selezionare in automatico le migliori impostazioni. Una soluzione particolarmente intelligente, soprattutto per coloro che non vogliono perdere troppo tempo nella gestione di queste voci che probabilmente interesseranno solo gli amministratori di sistema più esperti. Qui, è possibile specificare il livello di compressione, scegliere uno schema di autenticazione e il sistema di cifratura, nonché molteplici altre funzioni. Troviamo anche il server Xvnc che crea la propria sessione X per gli utenti remoti. Tuttavia, se volete connettervi a una sessione X esistente, potete utilizzare anche il server **x0vncserver.** Nella nuova versione, questo supporta anche l'estensione XDamage per il rilevamento delle modifiche.



› **VirtualGL project ha pubblicato un report molto dettagliato dopo la scoperta di vari problemi relativi alle prestazioni: anche i testi non sono perfetti...**

Performance

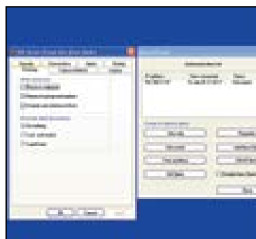
L'ultima versione di TigerVNC offre una serie di piccole migliorie per lo più nascoste a una prima occhiata. Oltre alle molteplici correzioni di bug, gli sviluppatori hanno ottenuto discreti risultati nel migliorare la gestione di client e server, nonché per garantire una più proficua implementazione di TLS nel client Java del progetto. Nei nostri test, TigerVNC ha lavorato sufficientemente bene. Sebbene sia possibile connettersi al server con altri client, consigliamo di farlo con il proprio, così da evitare possibili problemi. Sorprendentemente, poi, TigerVNC si è dimostrato molto più reattivo durante le sessioni Windows rispetto a quelle Linux. Con Gnome, poi, abbiamo rilevato un rallentamento anche troppo marcato. LXP



Caratteristiche



Codice pulito
Oltre alla correzione dei bug, gli sviluppatori si sono concentrati sull'ottimizzare il codice.



Cross-platform
TigerVNC può funzionare in modo ottimale anche su Windows o Mac OS.

LINUX PRO Giudizio

TigerVNC 1.4.0

Sviluppatore: TigerVNC Developers
Web: www.tigervnc.org
Licenza: GPL

Caratteristiche	5/10
Prestazioni	5/10
Facilità d'uso	5/10
Documentazione	5/10

» *Una versione che, seppur funzionando, non offre alcun beneficio convincente.*

Il voto di Linux Pro **5**

OpenPi

Una piccola scheda ricca di funzioni, i cui schemi sono totalmente Open Source e possono essere modificati da chiunque

In breve

» Una scheda basata su Pi Compute Module con transceiver wireless sub-GHz e un sensore termico. Gli schemi sono Open Source, così che chiunque dotato di competenze possa modificarli.

Specifiche

» Raspberry Pi Compute Module: Broadcom 2305 SoC 700 MHz, 512 MB di RAM, 4 GB di spazio

» OS: Raspbian

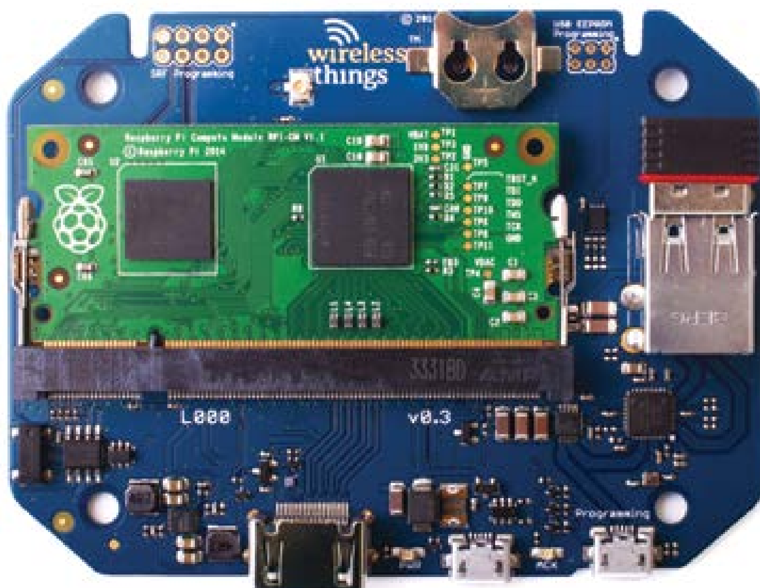
» Porte: 2x USB 2.0, HDMI, 2x micro USB

» RTC: TI BQ3200

» Sensore Temp: Ti TMP1000NA

» Sub-GHz Wireless: Wireless Things SRF

EVE è un dispositivo compatto che si adatta perfettamente sopra a un Raspberry Pi tramite i pin GPIO. Ci sono parecchi moduli che possono essere installati su una scheda EVE: SRF o RFM12B, orologi in tempo reale (RTC), sensori di temperatura e molto altro. Con **OpenPI**, Wireless Things fa un enorme balzo in avanti. Basta pensare che questo progetto, nel momento in cui scriviamo, è al centro di una serrata campagna di crowdfunding (**http://bit.ly/KickstartOpenPi**). Utilizzando il Raspberry Pi Compute Module come base, OpenPi fornisce due porte USB, un RTC (batteria inclusa), un transceiver SRF da 868-915 MHz (per le connessioni seriali a lungo raggio) e un sensore di temperatura. Tutto questo è confezionato in una custodia di plastica ordinata con il logo di Wireless Things. In termini di estensibilità, si possono ottenere 18 (più due di potenza) pin GPIO, le intestazioni per SRF e la programmazione USB EEPROM, così come i pin wire-by-hand per il collegamento a XBee di Wireless Things. Il Compute Module ha 4 GB di spazio su eMMC che consentono un'installazione standard di Raspbian. Per i makers più accaniti che non si accontentano di un



› *OpenPi* **è leggermente più grande di un Raspberry Pi. Gli angoli tagliati possono essere un omaggio a Battlestar Galactica**

comune Raspberry Pi, con l'aggiunta di OpenPi, si possono usare un trasmettitore e un RTC, più tutta un'altra serie di periferiche a costi ridotti. Magari, a vedersi, il Rasbperry Pi unito a OpenPi non sarà il massimo dell'ordine, ma basta ingegnarsi con una confezione di Lego per rendere decisamente più piacevole il quadro.
Un unico pacchetto OpenPi di base, che comprende solo il modulo Pi Compute e OpenPi PCB, costa 66 €.

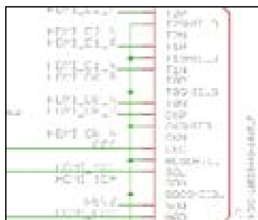
Hardware aperto

Ciò che rende unico OpenPI è che tutti gli schemi sono Open Source. Potete scaricarli come file EagleCad. Così facendo, OpenPi mira a rendere molto più facile utilizzare il Compute Module come una piattaforma flessibile su cui sviluppare progetti e prodotti. Le ricerche di marketing sostengono che la forma è importante tanto quanto le funzioni. Questo spiega in parte perché fino a ora OpenPi non sia decollato come piattaforma a uso comune. Infatti, almeno per adesso, trova un più largo uso in ambito industriale, dove viene preferito per le sue doti di versatilità e flessibilità nella progettazione. Infatti, se pensiamo anche al suo ambito di vendita, possiamo notare come gli sconti siano riservati soprattutto a chi fa acquisti in stock. Per esempio, abbiamo il 10% se ne ordiniamo più di 100, oppure il 30% se andiamo oltre le cinque cifre. LXP

Caratteristiche



Starter kit
Il Kit include un alimentatore da 5V, un cavo HDMI e una tastiera wireless con touchpad.



Schemi Open
Per i più curiosi, è possibile scaricare i circuiti in EagleCad e modificarli secondo le necessità.

LINUX PRO Giudizio

Wireless Things OpenPi

Sviluppatore: Wireless Things
Web: http://wirelessthings.net
Prezzo: 66 € circa

Caratteristiche	8/10
Prestazioni	7/10
Facilità d'uso	7/10
Qualità/prezzo	7/10

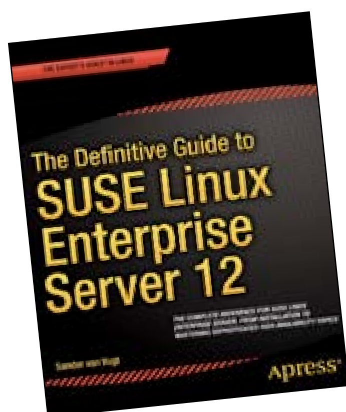
» *Un dispositivo che può stimolare la creatività, ma che per ora rimane appannaggio di ambienti industriali.*

Il voto di Linux Pro **7**

The Definitive Guide to SLES 12

Un valido aiuto per gli aspiranti sysadmin autodidatti

Le vie per diventare dei buoni amministratori di sistema sono infinite almeno quanto lo sono le distribuzioni GNU/Linux. Ma se volete lavorare in ambito enterprise, le scelte sono molto più limitate. Nel caso vi piaccia SUSE, la distribuzione che ha come mascotte un camaleonte verde, vi raccomandiamo decisamente questo volume. L'autore spiega i vari argomenti in modo decisamente chiaro e lineare, ma la lettura non è proprio una passeggiata, sia per il numero di pagine, sia per la complessità di molte delle soluzioni trattate. Nell'introduzione si afferma che è sufficiente una conoscenza base di GNU/Linux per comprendere quanto illustrato: secondo noi serve un po' di più, soprattutto nelle sezioni avanzate. In tutti i casi i primi capitoli sono pensati proprio per spiegare gli argomenti base, dall'installazione di SLES 12 alla gestione degli utenti. A partire dal capitolo 6 si inizia a far sul serio, però: sicurezza, networking, script di shell, virtualizzazione, cluster e, infine, gestione di uno o più sistemi SLES tramite SUSE Manager, un tool davvero ben fatto. Il tutto è condito con varie schermate, un buon numero di comandi d'esempio e, cosa che abbiamo davvero apprezzato, diversi esercizi da risolvere per mettersi alla prova e capire veramente se e quanto si è imparato dalla lettura. Per seguire gli esercizi, così come il testo, conviene avere sottomano una macchina con SLES installata (la distro in sé, essendo Open Source, è scaricabile liberamente): riteniamo che solo così sia possibile imparare a fondo quanto spiegato. L'unica vera pecca è il prezzo, che ci sembra un po' troppo alto nonostante la bontà dei contenuti, davvero ricchi e ben spiegati. LXP



LINUX PRO Giudizio

The Definitive Guide to SLES 12

Autore: Sander van Vugtć
Editore: Apress
ISBN: 978-1-4302-6821-5
Prezzo: 99,99 $ (stampato), 69,99 $ (ebook)
Pagine: 568

» *Ricchissimo di informazioni ben spiegate, ottima la presenza di esercizi da svolgere, piuttosto costoso.*

Il voto di Linux Pro 8

Siti efficaci con Joomla!

Ottimizzare e portare al successo il vostro sito Joomla

Creare un sito Web con un CMS è un'operazione alla portata di tutti. Per chi non lo sapesse, ricordiamo che un CMS è un software Web per la creazione di pagine Internet dinamiche senza necessariamente dover mettere mano all'HTML o a qualche altro linguaggio di programmazione. Quindi, è facile, abbiamo detto, ma da qui a realizzare un sito che sia sicuro, efficace, veloce e magari anche facilmente indicizzabile dai motori di ricerca ce ne vuole. Claudio Romeo ha quindi deciso di scrivere il volume **Siti efficaci con Joomla!** proprio per spiegare queste cose. Forte di una lunga esperienza con Joomla! e della collaborazione con l'associazione Joomla! Lombardia, Claudio ha raccolto un buon numero di consigli e "best practice" su svariati argomenti. Premettiamo che non si tratta di un libro per principianti: la trattazione prevede che si sia in grado di impostare un sito autonomamente, quindi niente spiegazioni sull'installazione base del CMS, tanto per capirci. I capitoli trattano, in successione, modifica dei template, sicurezza, ottimizzazione delle prestazioni, uso dei font, gestione multilingua, indicizzazione da parte dei motori di ricerca, per concludersi con la descrizione delle estensioni fondamentali e del troubleshooting. Claudio è riuscito a trattare argomenti a volte anche impegnativi in modo estremamente scorrevole e chiaro, inserendo anche numerose immagini per illustrare quanto detto a parole. Tra i capitoli che più ci sono piaciuti, dobbiamo citare il terzo, quello sulla sicurezza, e il quarto, sull'ottimizzazione delle prestazioni. In entrambi i casi si trovano dei pratici suggerimenti su come comportarsi per raggiungere lo scopo, senza dover fare chissà quali salti mortali. Infine una nota gradevole: Claudio, da sempre impegnato nel diffondere l'Open Source, ha interamente realizzato questo volume usando Scribus e altri tool liberi, complimenti! LXP



LINUX PRO Giudizio

Siti efficaci con Joomla!

Autore: Claudio Romeo
Editore: Tecniche nuove
ISBN: 978-88-481-2978-7
Prezzo: 23,90 € (stampato), 22,00 € (ebook)
Pagine: 232

» *Quanto scritto in copertina è proprio vero: dopo aver letto il libro si ha il controllo completo sul proprio CMS.*

Il voto di Linux Pro 9

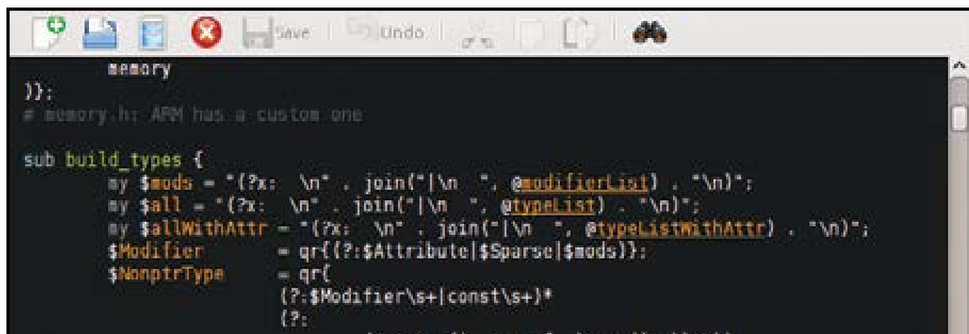
Il confronto

» Ogni mese mettiamo a confronto prodotti e programmi per farvi scegliere al meglio!

Linguaggi di scripting

Linux Pro vi porta oltre Bash, esplorando diversi linguaggi di scripting per scoprire qual è il migliore e il più adatto agli amministratori di sistema





Modalità del test

Come sempre, è importante premettere che questo confronto, così come gli altri che si sono succeduti e si succederanno sulle pagine di Linux Pro, è frutto di prove, ma anche di opinioni collettive che talvolta portano a preferire questa o quell'altra soluzione. In altri termini, non c'è un linguaggio di scripting migliore, quanto più una scelta oculata e fatta in base alle nostre necessità. Un linguaggio di scripting viene giudicato dagli amministratori di sistema non in base a quanto facilmente riesce ad aiutarli a risolvere i problemi, ma secondo quanto tempo fa risparmiare nella scrittura degli script per via di quante soluzioni sono già state scritte e disponibili per il download, magari correlate da istruzioni ben dettagliate su come adattarle ai vari casi che si pongono. Abbiamo quindi scelto dei linguaggi che coprissero la gamma di versioni installate in genere in una rete.

Ogni amministratore di sistema, pur di risparmiare tempo, ama crearsi delle scorciatoie con tutta una serie di script preimpostati. Di solito, ogni nuovo Sysadmin assunto in qualche azienda chiede subito quale linguaggio di scripting viene usato e all'occorrenza lo studia, lo impara e poi lo applica quando meglio crede. Chi ha esperienza con più linguaggi, sa che non ne esiste uno migliore dell'altro. In questi termini, è meglio parlare di piattaforma più adatta in base al contesto in cui ci troviamo. La maggior parte dei linguaggi è sulla cresta dell'onda da parecchio tempo. Basta pensare a newLISP che è stato avviato su una workstation Sun-4 nel 1991. Nel corso del tempo, i vari ambienti si sono poi influenzati tra loro, ampliando le librerie e generando soluzioni alternative nel momento in cui si trovavano di fronte a problemi di varia natura. Per scegliere un linguaggio di scripting, anziché puntare su quanto velocemente può essere eseguito lo script creato si tende a valutare la rapidità e la semplicità di scrittura. In definitiva, quindi, si sceglie guardando alle potenzialità che un dato linguaggio offre per migliorare l'esperienza d'uso di Linux e dei sistemi di rete. Ed è proprio in tale ottica che abbiamo affrontato questo confronto serrato.

> **"Di solito i Sysadmin neo assunti chiedono quale linguaggio di scripting devono imparare"**

La nostra selezione

- » Bash
- » Perl 5
- » Python
- » Ruby
- » NewLISP

Curva di apprendimento

Risorse online, libri e documentazione

Le prime domande da porsi sono due: il linguaggio è facile da imparare? Le risorse per apprenderlo sono sufficienti allo scopo? Se anche questi due quesiti hanno una risposta positiva, c'è un altro fattore da valutare: la solidità e la presenza di una comunità attiva. Per produrre script di backup e di prova in ciascun linguaggio, abbiamo iniziato navigando in Stack Overflow. Scaricando codice casuale, però, non abbiamo riscontrato alcuna coerenza tra Posix, Bash e Legacy. Fortunatamente, però, **www.shellcheck.net** ci ha aiutato non poco a valutare la correttezza degli script. In più, è utile per imparare le migliori pratiche di correzione. Il **Linux Document Project** comprensivo di **Advanced Bash Scripting Guide** (**www.tldp.org/LDP/abs/html**) è eccellente per aiutarvi a prendere confidenza con **Bash**. Passando a **Perl**, non si può fare a meno di lodare la quantità di documentazione online presente, che a tratti viene perfino definita leggendaria. Ciò nonostante, almeno nel nostro caso, abbiamo deciso di iniziare con un semplice esercizio del classico libro per admin di O'Reilly. Se però non siete avvezzi ai tomi, consigliamo di dare un'occhiata a **http://perlmonks.org**. Per quanto riguarda **Python**, invece, si tratta di uno dei linguaggi di scripting più aperti in circolazione, tanto da renderlo un ottimo punto di partenza per i neofiti che si avvicinano per la prima volta alla programmazione di script. Tuttavia, a causa dei tanti sotto-processi che persistono per ragioni di compatibilità, è necessario porre molta attenzione alla lettura del codice. Per questo consigliamo di partire con Python 3 anziché Python 2. Per una vasta quantità di guide e tutorial, potete consultare **www.python.org/about/gettingstarted**. **Ruby** è un altro linguaggio di scripting adatto a chi vuole studiare e avvicinarsi a questo genere di attività. Il libro di David B. Copeland, **Build Awesome Command Line Applications in Ruby** vi farà risparmiare ore di apprendimento online. Infine, arrivando a **newsLISP**, possiamo dire che si tratta di un linguaggio piuttosto complesso da imparare. Una volta prese le redini, però, si dimostra molto versatile. Basta vedere cosa riesce a fare con il solo uso di elenchi, funzioni e simboli. Per iniziare, consultate il sito **http://newlisp.org**.

In code that I show you to see what happens. Play around with it and make changes. The worst that can happen is that it won't work. When I type in code it will be formatted like this:

```
##Python is easy to learn
print("Hello, World!")
```

› **Per imparare Python, basta solo aver voglia di cercare. Le risorse sono dappertutto, tanto sulla carta quanto online**

Verdetto

Bash ★★★★☆
NewLISP ★★★★☆
Perl 5 ★★★★☆
Python ★★★★★
Ruby ★★★★★

» *Python e Ruby, visto il materiale, sono i più facili da imparare.*

Versione e compatibilità

Quanto sono versatili e pratici da usare?

La domanda da porsi è: stiamo usando la versione corretta? Iniziamo con Bash. Questo linguaggio di scripting è stato rilasciato nella release 4 ben sei anni fa. La versione 4.2 ha aggiunto poco alla base più che corposa e anche dopo quattro anni, quando è stato rilasciato Bash 4.3, i cambiamenti sono stati molto lievi. Perl viene incluso nella maggior parte delle distribuzioni. L'ultima versione è la 5.20.2, ma molte distro stabili lavorano già con la 5.18. I miglioramenti sono minimali e qualsiasi script vogliate scrivere farà perfettamente il proprio lavoro. Come abbiamo già detto, per quanto riguarda Phyton è meglio partire dalla versione 3. Noi abbiamo utilizzato come base di riferimento la 3.3, in quanto la 3.4 non ha comunque aggiunto alcuna nuova funzione di sintassi. Ruby, sotto il profilo dell'avvicendamento delle versioni, ha una storia un po' più complessa. A ogni release, infatti, si sono verificati dei problemi che hanno richiesto soluzioni talvolta anche creative. Rvm, comunque, consente di eseguire più release di Ruby contemporaneamente. newLISP vanta una stabilità sorprendente, ma purtroppo ha un difetto: non tutti gli script possono essere eseguiti con le ultime versioni disponibili.

```
print "\n${l_hlt}" . " "x${l_columns} . "${l_rst}\r"
    . "${l_hlt}  ${l_int}BRANCH:${l_rst}${l_hlt} ${l_git_branch}  (${l_git_sha
}  ${l_rst}\n\$ ";
EOF

}

export COLUMNS
PS1="${PS1}\$(Func_GitCheck)"
Func_GitCheck
richard@luggable:~/work$ bash --version
bash --version
GNU bash, version 4.3.11(1)-release (i686-pc-linux-gnu)
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Func_GitCheck
richard@luggable:~/work$
```

› **Se utilizzate le più recenti versioni di Bash, vi accorgerete che sono state aggiunte diverse migliorie, molte delle quali inaspettate**

Verdetto

Bash ★★★★★
newLISP ★★★★☆
Perl 5 ★★★★☆
Python ★★★☆☆
Ruby ★★★★☆

» *Ruby è versatile, ma Bash non mostra alcun problema.*

Scripting rivolto al Web

Spostate gli script in ambiente HTTP

Gran parte della vita di un Sysadmin si svolge sul Web. I linguaggi di scripting devono quindi tenere il passo con i tempi. Abbiamo esaminato sia la semplicità con cui si scrive il codice, sia la ricerca di soluzioni disponibili per compiere qualsiasi azione. Uno dei difetti di questi linguaggi è la differenza nell'uso delle espressioni e dello stile per produrre risultati simili. Tuttavia, si tratta di un fattore che può passare in secondo piano rispetto al supporto per il funzionamento in locale. Ruby è veloce, Python leggibile e newLISP potente. Tutto però rimane relativo, senza un ambiente favorevole allo sviluppo del codice per le proprie reti.

Bash ★★☆☆☆

Sicuramente Bash non sarà la prima scelta per chi cerca un linguaggio di programmazione rivolto al Web. Tuttavia è bene sapere che quando il server non supporta il principale linguaggio di scripting per cui avete optato, si può ripiegare utilizzando **bashlib**. Questo script ha il compito di rendere più tollerabile la programmazione CGI in una shell Bash. Lo script in questione sarà pieno di echo e intervalli dovuti all'introduzione dei comandi per produrre l'output desiderato. Inoltre, volendo fare qualche considerazione in merito alla sicurezza, ve ne sconsigliamo l'uso in ambiente Internet aperto. Tuttavia, a dispetto di tutto questo, vale la pena ricordare che Bash funziona molto bene come linguaggio di prototipazione. Infatti si possono riempire interi file di testo con commenti che descrivono la struttura che si desidera.

```
archivedownload.sh (~/lud) - gedit
File Edit View Search Tools Documents Help
bak-test.sh ×   archivedownload.sh ×
if ls -U * text.pdf > /dev/null 2>&1; then
  echo Found text-format PDFs, moving into text/ directory...
  if [ -d text ]; then
    mv * text.pdf text/
  else
    mkdir text
    mv * text.pdf text/
  fi
  echo Move complete.
fi
echo Deleting temporary files...
rm identifiers.txt processedidentifiers.txt
            sh ▾      Tab Width: 8 ▾      Ln 34, Col 44
```

```
Now enter the URL (minus the http://) of the blog (eg: newlisponrockets.cc
gonetoearth.org
Now enter the owner of the blog (eg: Rocket Man)
Richard Smedley
Now setting up Posts, Users, and Comments tables...
Enter a database name (.db extension added automatically): gonetoearth
Enter a user name for the ADMIN user (case sensitive): garddwyr
Enter an email for the ADMIN user (case sensitive): root@localhost
Now enter a password for the ADMIN user (case sensitive): admin123
Salt: 5B3B1D8C-1303-4DD8-ADFB-5AAD44D0037C
Cookie Salt: 55A5FDC8-63A8-4CA0-835E-0AD80AF7647C
Password hash: ac9565714f4c48a0185e00720522c1413bdf65d8

true
true
User data: ((0 "root@localhost" "ac9565714f4c48a0185e00720522c1413bdf65d8"
3-4DD8-ADFB-5AAD44D0037C"
  0 nil nil "garddwyr" "55A5FDC8-63A8-4CA0-835E-0AD80AF7647C" nil nil nil

root@luggable:/var/www/newlisp#
root@luggable: /var/www/new
```

newLISP ★★★★☆

Code Patterns, scritto dal creatore di newLISP Lutz Mueller, è disponibile sul sito **www.newlisp.org** e contiene capitoli riguardanti HTTPD e CGI, così come TCP/IP e UDP. Avrete quindi tutto ciò che serve per iniziare. La connettività di rete integrata in newLISP e la sintassi piuttosto semplice vi renderanno facile generare pagine HTML con svariati script. Per usare un framework già pronto, potete sfruttare newLISP on Rockets che utilizza Bootstrap, jQuery e SQLite, combinando la rapidità di sviluppo con ottimi risultati finali. newLISP on Rockets, inoltre, fornisce diversi operatori: **from** (convert-json-to-list), via (twitter-search) e **to** (display-post-box) che vi aiuteranno ad aggiungere diverse funzionalità Web. Purtroppo, a tutto questo si affianca una comunità ancora ristretta e che stenta a far decollare il ritmo di sviluppo a oggi piuttosto lento.

Comunità di supporto

Ci sono comunità abbastanza grandi da sostenere i progetti?

La quantità di risorse e servizi a disposizione di un Sysadmin cresce esponenzialmente. Tuttavia c'è bisogno di un costante apprendimento. Tutti utilizziamo Bash, ma è comunque necessario imparare almeno un altro linguaggio di scripting. Perl è stato per anni uno dei più utilizzati, soprattutto nel decennio tra il 1980 e il 1990. Lentamente ha però perso terreno rispetto a Python e a Ruby, ma chi ha iniziato a lavorare negli anni '90 non farà fatica a trovare qualcuno che lo aiuti con gli script in Perl. Python, invece, è diventato uno dei linguaggi nevralgici del nuovo mondo basato su Internet, tanto che Google ne fa largo utilizzo. Gran parte del software utilizzato dai Sysadmin, infatti, è basato su Python, anche se può essere supportato anche da Ruby. Quest'ultimo ha il vantaggio di essere alla base di Chef e Puppet, così come Vagrant e Travis CI. Questo significa che la sua diffusione è piuttosto marcata e non ci saranno problemi a trovare qualcuno che se ne intenda. newLISP, infine, conta su una comunità piuttosto ristretta e purtroppo, almeno per adesso, non mette a disposizione molte soluzioni già pronte. Certo, il buon supporto della Community online aiuta, ma non abbastanza.

Verdetto

Bash ★★★☆☆
NewLISP ★★☆☆☆
Perl 5 ★★★★☆
Python ★★★★☆
Ruby ★★★★★

» *Ruby è tra i linguaggi più diffusi e con una comunità attiva.*

Perl 5 ★★★★☆

Perl è stato il primo linguaggio di scripting Web CGI e più o meno è riuscito sempre a tenere il passo con i tempi. Nonostante la presenza di librerie e di una buona documentazione è però un linguaggio di scripting da scegliere con cura. Catalyst, Dancer e Mojolicious sono tutti ottimi framework Web e molto più probabilmente troverete ciò che vi serve in CPAN. È possibile incollare insieme alcune librerie, molte delle quali sono già presenti nelle distro, così da gestire una pipeline di compiti come il recupero di informazioni XML, la conversione dei dati in file PDF e l'indicizzazione di una pagina Web. La tradizionale interfaccia CGI di Perl è ancora oggi disponibile. Nonostante la possibilità di utilizzare alternative più potenti in PSGI, l'uso di CGI è ancora consigliato.

```
if ($q->param()) {
    # Parameters are defined, therefore the form has been submitted
    display_results($q);
} else {
    # We're here for the first time, display the form
    output_form($q);
}

# Output footer and end html
output_end($q);

exit 0;

#-------------------------------------------------------------

sub output_top {
    my ($q) = @_;
    print $q->start_html(
        -title => 'Back-up selection',
```

```
            filename, _, _ = related_attachment
            content = re.sub(r'(?<!cid:)%s' % re.escape(fil
%s' % filename, content)
        self.alternatives[i] = (content, mimetype)

    return super(EmailMultiRelated, self)._create_alternatives(msg)

def _create_related_attachments(self, msg):
    encoding = self.encoding or settings.DEFAULT_CHARSET
    if self.related_attachments:
        body_msg = msg
        msg = SafeMIMEMultipart(_subtype=self.related_subtype, enco
        if self.body:
            msg.attach(body_msg)
        for related_attachment in self.related_attachments:
            if isinstance(related_attachment, MIMEBase):
                msg.attach(related_attachment)
            else:
* backup-reporter.py:61:0 Python
```

Python ★★★★★

PythonWeb Server Gateway Interface (WSGI) gestisce l'interfaccia Web server, mentre le librerie WSGI si occupano della gestione delle sessioni, dell'autenticazione e di qualsiasi altro problema che si vuole affrontare. Python possiede anche parecchi framework Web full-stack, come Django, TurboGears e Pylons. In modo simile a Rails, per alcuni scopi potrebbe essere meglio codificare le funzionalità Web su uno script esistente, ma in compenso potrete evitare di creare un pasticcio di HTML misto a Python. Questo linguaggio di scripting ha molti altri vantaggi. Tra questi è possibile annoverare il proprio interprete che funziona con qualsiasi framework per applicazioni compatibili con WSGI. In più, consente di provare programmi scalabili in un ambiente pulito e funzionale.

Ruby ★★★★★

Rails non deve essere considerato come una panacea per i problemi. **Sinatra**, inoltre, pur rendendo certamente facile scrivere in Ruby per il Web, spesso si dimostra eccessivo per la maggior parte degli scopi. Premesso questo, Rails fa il suo lavoro nel permettervi di ottenere codice ben sviluppato senza annegare nella sintassi. Ruby è l'ideale per ottenere uno script *Web-enabled*, grazie soprattutto a chi nel tempo lo ha sviluppato in modo organico e ben organizzato. Aggiungere un'interfaccia Web al nostro script di backup, oltre che divertente, è stato veloce. Strumenti come **nanoc** che genera HTML statico da HAMS completano l'espressività del linguaggio e lo rendono ideale da utilizzare per qualsiasi funzionalità.

```
log:
development.log

public:
404.html  422.html  500.html  favicon.ico  robots.txt

test:
controllers  fixtures  helpers  integration  mailers  models  t

tmp:
cache  pids  sessions  sockets

vendor:
assets
richard@luggable:~/work/code/ruby/rails/crash/code/blog$
[richard@luggable: ~/work/co]
```

Programmabilità

Gestire grandi script richiede esercizi di programmazione

Prima di arrivare a 1.000 righe di codice, gli script in Bash diventano ingestibili. Vista la sua natura procedurale, i tentativi per creare un Bash *object-oriented* (OO) non mancano. Ciò nonostante, vi sconsigliamo questo genere di approccio, pensando sia molto meglio procedere per vie modulari. Infatti, il *Functional Programming* (FP) in Bash è impraticabile (**http://bit.ly/BashFunsh**). Perl, invece, gestisce piuttosto bene il FP, anche se non è un'operazione adatta a tutti. Le funzioni sono oggetti di prima classe, ma purtroppo alcune caratteristiche mancano. Ruby è concepito come linguaggio OO puro ed è forse il migliore nel suo genere. Può anche essere utilizzato con uno stile funzionale. Per ottenere il codice FP di Ruby, però, dovrete sviare così tanto dalle basi che in definitiva finirete per imparare un altro linguaggio. Infine passiamo a newLISP che si dimostra elegante e potente con tutte le caratteristiche funzionali a portata di mano. newLISP, infatti, utilizza un'implementazione pseudo OO che consente di dedicarsi al *functional-object oriented programming* (FOOP). Tuttavia, non significa che sia possibile avvicinarsi bene alla programmazione OO reale.

Verdetto

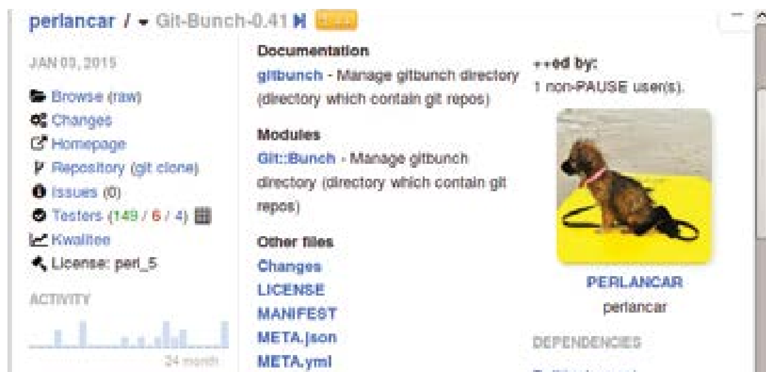
Bash ★★☆☆☆
NewLISP ★★★☆☆
Perl 5 ★★★☆☆
Python ★★★★★
Ruby ★★★★

» *Python è un linguaggio multi paradigma ed è il più facile da gestire.*

Estendere i linguaggi

Librerie, moduli e molto altro ancora

Nessuno di questi linguaggi di scripting è così ricco di classi da ricorrere a librerie *non-core* (o moduli come talvolta vengono chiamati). In base alla loro completezza, però, la semplicità con cui si può gestire uno script varia notevolmente. Perl continua a stupire per gran parte delle sue funzioni, ma il suo approccio unilaterale ai problemi, che implica l'impossibilità di risolverli con diverse soluzioni, può lasciare sopraffatti. Meno evidente è la disponibilità di estensioni da parte di Bash, che tuttavia sono state create per risolvere malfunzionamenti di varia natura. Python ha un'eccellente supporto di librerie. La comunità, inoltre, prima di aggiungere qualsiasi elemento esterno, lo valuta con molta attenzione, cercando di limitare al minimo gli eventuali problemi di compatibilità. La preoccupazione di "fare la cosa giusta" è alla base della filosofia decisionale degli sviluppatori di Python. Tuttavia, le soluzioni alternative sono a portata di mano. Basta pensare alla piena adozione del gestore pacchetti Pip con Python 3.4 che ha assicurato la parità con Ruby. RubyGems fornisce il formato di distribuzione perfetto per le librerie Ruby e i vari programmi. In più, Bundler gestisce il tutto per ottenere dipendenze e versioni corrette. Il vostro unico problema sarà trovare una guida che descriva bene la proliferazione delle librerie di Ruby. newLISP non è un grande linguaggio, ma non ha bisogno di molti moduli esterni per fare bene il suo lavoro. Ciò nonostante, le librerie di base non mancano.



› *CPAN* **è una delle migliori e più utili risorse per Perl**

Verdetto

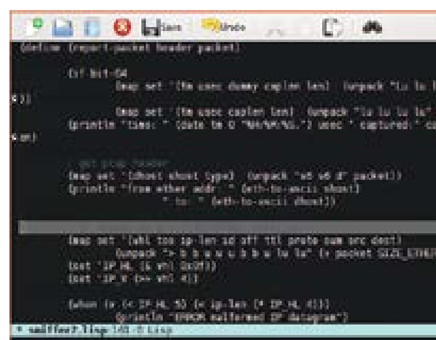
Bash
★★★☆☆
NewLISP
★★★☆☆
Perl 5
★★★★★
Python
★★★★☆
Ruby
★★★★☆

» *La popolarità di CPAN si sposa bene con una notevole organizzazione.*

Sicurezza di rete

Test e sicurezza dei network locali esposti al pubblico

I test di penetrazione e gli esami di forensica informatica sono all'ordine del giorno per i Sysadmin responsabili di un'infrastruttura di rete medio-grande. La buona notizia è che gli strumenti per eseguire queste operazioni sono presenti e possono essere inclusi in un unico script di shell. Partendo da Bash, però, possiamo dire che questo linguaggio non è il più adatto a questo genere di operazioni. Perl, dal canto suo, ha visto un brusco calo di utilizzo nel campo della sicurezza dopo l'avvento di Metasploit. Tuttavia gli strumenti ci sono ancora e vengono supportati da una vasta comunità che non pensa lontanamente a migrare verso altri linguaggi. Perl ha strumenti come **Pweb**, una raccolta di funzioni dedicata alla sicurezza delle applicazioni Web e alla verifica delle vulnerabilità. Lo troviamo incluso in alcune distribuzioni, come Kali e Backbox. Strumenti come **Wireshark** sono potenti e consentono di ispezionare i pacchetti, anche se a volte sentirete il bisogno di andare oltre. Python, a questo proposito, non ha solo **Scapy**, ma fornisce anche una libreria socket per leggere e scrivere facilmente i packet. I blocchi di Ruby (funzioni di scrittura *on-the-fly*) e altre caratteristiche sono l'ideale per la scrittura di codice di rete asincrono, nonché per la prototipazione rapida Python. Il più grande vantaggio di Ruby, però, sta in **Metasploit** che è anche il software di pen-test più utilizzato. In definitiva, in termini di strumenti pronti, potete fare un mix tra Perl, Python e Ruby. Questi tre forniscono tutto il necessario per esaminare rapidamente una rete e trovarne le debolezze. Infine parliamo di newLISP, che sebbene non sia tra i più noti a livello di sicurezza informatica, consente ugualmente di creare pacchetti grezzi per i test principali. Ancora una volta, quindi, ci troviamo di fronte a un linguaggio con del potenziale, ma che rimane limitato a causa del ristretto numero di utenti su cui può contare.



› *newLISP*, **pur avendo pochi strumenti, ha un gran numero di funzioni di rete**

Verdetto

Bash
★☆☆☆☆
NewLISP
★★★☆☆
Perl 5
★★★★☆
Python
★★★★★
Ruby
★★★★★

» *Python è davanti a Ruby e Perl, ma tutti sono amici della sicurezza.*

Linguaggi di scripting

Il verdetto

Ammettiamo che stilare un verdetto, in questo caso, è piuttosto difficile. Bash, per esempio, pur non essendo il migliore linguaggio di scripting, ci fa risparmiare tempo e può essere gestito meglio degli altri quattro. In altre parole, conoscere questo linguaggio è piuttosto essenziale e niente dovrebbe farvi rinunciare al piacere di studiarne alcune sue parti. Perl viene immediatamente dopo ed è il passo successivo da compiere dopo aver appreso Bash. Certo, soffre un po' il confronto con gli altri, ma continua a essere il coltellino svizzero della CLI di Linux. In più, Perl può contare su una comunità di supporto enorme. newLISP si è rivelato una piacevole sorpresa. Sì, ci sono delle limitazioni, come la comunità ridotta che non riesce a sviluppare a pieno la forza del linguaggio, ma le potenzialità non mancano e basta solo che qualcuno inizi a sfruttarle al meglio. Python è potente ed è un ottimo ambiente multi-paradigma. La comunità, inoltre, è abbastanza grande e disponibile nell'aiutarvi a risolvere la maggior parte dei problemi. Per molti versi, Python può perfino essere definito come un linguaggio didattico, insegnato sempre più spesso nelle scuole con percorsi formativi più o meno basilari. In più, consente di creare un ottimo bagaglio di conoscenze per i SysAdmin impegnati nell'ambito della sicurezza. Seppure non riesca a rendere banali le situazioni più difficili, evita per lo meno di farvi venire il mal di testa. Rails ha

```
private

def vm_host(vm)
  host_options = {
      user: vm['user'] v 'vagrant',
      hostname: vm['hostname'] v 'localhost',
      port: vm['port'] v '22',
      password: vm['password'] v 'vagrant'
  }

  SSHKit::Host.new(host_options)
 end
end
end
```

› **È impossibile non riconoscere il fascino di Ruby**

contribuito ad accendere i riflettori su Ruby, ma Chef, Puppet e Vagrant hanno rivelato cosa si può realmente fare con il linguaggio di scripting espressivo sviluppato da Yukihiro Matsumoto. In definitiva, abbiamo deciso di mettere Ruby sul primo posto del podio, lasciando a Python il secondo e Perl il terzo. Ciò nonostante, è importante sottolineare come per i Sysadmin sia importante avere un'infarinatura di tutti i linguaggi di scripting di cui abbiamo parlato.

1° Ruby



Web: www.ruby-lang.org **Licenza:** GPLv2 o 2-clause **Versione:** 2.2.1

» Potente, espressivo e molto veloce da imparare.

2° Python



Web: www.python.org **Licenza:** PSFL **Versione:** 3.5.0a3

» Multiparadigma, incoraggia le buone pratiche e ha una buona comunità.

3° Perl 5



Web: perl.org **Licenza:** GPL o Artistic License **Versione:** 5.20.2

» Ancora uno dei più grandi amici dei Sysadmin.

4° newLISP



Web: www.newlisp.org **Licenza:** GPL **Versione:** 10.6.2

» Ha delle potenzialità, ma non viene ancora sfruttato a pieno.

5° Bash



Web: www.gnu.org/software/bash **Licenza:** GPLv3+ **Versione:** 4.3.30

» Non è adatto a tutto ma è ancora essenziale.

A voi la parola...

Non siete d'accordo con le nostre scelte? Avreste usato altri linguaggi? Inviate le vostre opinioni su questo confronto a: **recensioni@linuxpro.it**

Considerate anche...

Sebbene Bash sia essenziale in alcuni campi, per lo scripting di shell tradizionale potrebbe essere utile guardare anche a **Zsh**. Quest'ultimo può contare su alcune piccole differenze e migliorie, come l'accesso alle variabili di posizione e la possibilità di estendere la shell tramite le funzioni di widget. Un'ottima alternativa ai linguaggi di cui abbiamo parlato può essere **Rebol** (*Relative Expression Based Object Language*) che tuttavia ha saltato il guado verso l'Open Source solo un paio di anni fa. Forse un po' troppo tardi per diventare un fenomeno di massa. Ciò nonostante, Rebol può contare su un design elegante e su una buona sintassi. In più è particolarmente utile nel trattare lo scambio e l'interpretazione delle informazioni tra sistemi informatici distribuiti, ma consente anche di realizzare potenti script di shell in modo piuttosto conciso. In definitiva, se siete alla ricerca di un nuovo linguaggio di scripting da imparare per il 2015, Rebol potrebbe proprio fare al caso vostro. LXP

Da non perdere

Fifth » Xfdesktop » GNU Hello » QuiteRSS » Audex » XDecorations » Debreate » Chocolate Doom » VoltAir » Budgie Desktop » ClamAV

Web browser

Fifth

Versione: 0.2 **Web:** http://fifth-browser.sourceforge.net

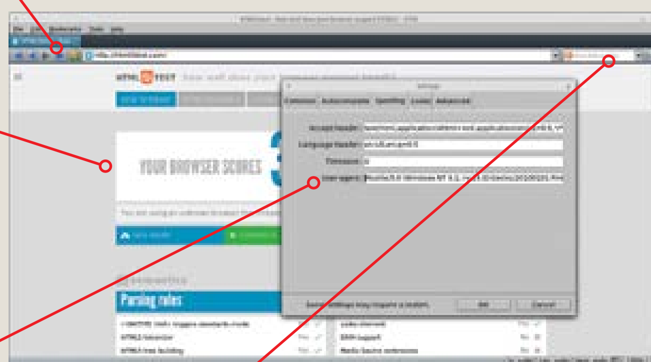
A dire il vero, non ci aspettavamo che il rilascio di Opera 12.x producesse una serie di reazioni così marcate sulle sue controparti Open Source. Infatti, a seguito di questo aggiornamento, i browser liberi basati sulla filosofia alla base di Opera si sono moltiplicati. Uno dei tanti che abbiamo deciso di prendere in considerazione in questo "Da non Perdere" è proprio **Fifth**. Si tratta di un browser basato su una combinazione abbastanza rara di motori FLTK toolkit e Webkit. I suoi sviluppatori dichiarano che Fifth è veloce come la luce e da parte nostra non c'è alcun intento a sconfermare questa affermazione. Infatti, provandolo, abbiamo constatato come anche le pagine più ricche di contenuti si aprano in pochi istanti. La versione 0.1.x può essere definita un punto di partenza o un trampolino per funzioni che verranno sviluppate a seguire. Infatti, il sito Web del browser, sotto questo aspetto, promette grandi innovazioni che verranno aggiunte mano a mano che si procederà con le prossime release. L'idea alla base di Fifth è il minimalismo. Il browser non supporterà Flash o WebGL, né sarà richiesto di utilizzare estensioni JavaScript. Al contrario, Fifth prende il controllo di tutto ciò che incontra sul Web, impedendo ai siti che effettuano lo spoofing di carpire i dati utili al vostro rilevamento. La prova sul campo di questo browser è positiva oltre qualsiasi ragionevole dubbio.
Si dimostra veloce e ha un'interfaccia pratica e ordinata che ricorda da vicino le prime versioni di Opera.
Le caratteristiche fornite sono solo quelle essenziali: schede, segnalibri, cronologia, download manager, password saver, selezione rapida e poco altro. Nel nostro test, Fifth ha raggiunto il punteggio tutto sommato modesto di soli 345 punti. Va però detto che è stato rilevato come "browser sconosciuto". Un elemento, questo, che può rendere orgogliosi i suoi sviluppatori. Attualmente, tutte le sue caratteristiche sono implementate in codice C nativo o C++, così da mantenere prestazioni elevate con un consumo di memoria minimo. Fifth, infatti, occupa solo 3,2 MB e può essere considerata una soluzione ideale per macchine con poche risorse.



› Fifth evita lo spoofing da parte dei siti Web

> **"Fifth è uno dei migliori browser per proteggere la vostra privacy"**

L'interfaccia di Fifth



Strumenti semplici
Fifth ha un look molto classico che unisce la semplicità della grafica alla funzionalità degli strumenti a disposizione.

Webkit rendering
Fifth, grazie a Webkit, sfrutta un motore di rendering molto veloce. Se i vostri siti Web preferiti non dipendono da plug-in di terze parti, non potrete chiedere di meglio.

Niente spoofing
Lo User Agent di Fifth non permette la rilevazione dei dati da parte dei siti Web. Infatti, nella nostra prova, è stato segnalato come "Browser sconosciuto".

Controllare la privacy
Personalizzate quello che preferite, ma non modificate il motore di ricerca. DuckDuckGo, infatti, preserva la vostra privacy.

Comandi veloci
Questi minuscoli controlli consentono di attivare o disattivare il supporto per JavaScript o CSS, così da visualizzare eventuali siti correttamente.

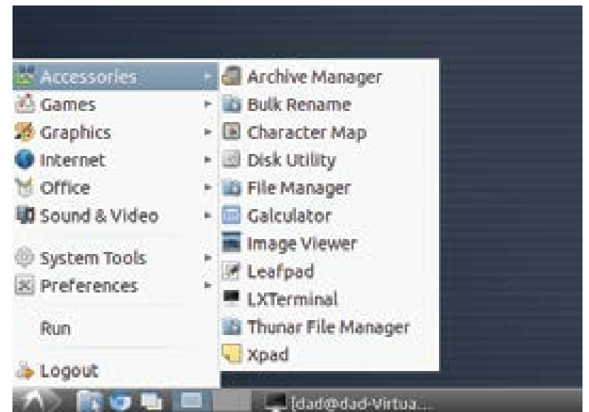
Desktop manager

Xfdesktop

Versione: 4.10.3 **Web:** http://bit.ly/Xfdesktop

Il nome e il numero di versione sono sufficienti per capire a dovere con cosa abbiamo a che fare. In pratica, dopo un lungo ritardo, gli sviluppatori di Xfce hanno deciso di pubblicare la nuova versione stabile di **Xfdesktop**, il desktop manager dell'ambiente Xfce. Quest'ultimo è noto per essere un ambiente tradizionale e leggero, ricco di funzioni e con una serie di strumenti che ne rendono l'utilizzo pratico e intuitivo. Uno dei suoi principali svantaggi è stato ed è tutt'ora uno sviluppo piuttosto lento, tanto che molti suoi estimatori avevano dato il progetto per abbandonato. Questa nuova versione, però, sembra sconfermare queste supposizioni, lasciando presagire una vita ancora piuttosto lunga. Secondo alcuni, tuttavia, le fasi di sviluppo di questo desktop sono da definirsi stagnanti. Per altri, invece, pur riconoscendo la lentezza con cui le nuove release si affacciano al pubblico, percepiscono questi ritardi come un elemento positivo. Infatti, Xfce implementa un paradygm molto stabile, che consente al desktop di funzionare in modo egregio e senza problemi. Un aspetto, questo, riconosciuto dai più come essenziale. Xfdesktop è uno dei componenti essenziali di Xfce. Gestisce le icone, il menu che si apre con il tasto destro del mouse e le impostazioni del desktop. La nuova versione porta con sé una serie di miglioramenti, tra cui la possibilità di verificare la presenza di thumbnail in nuove posizioni, una gestione del cestino più stabile e molto altro ancora. Il pacchetto Xfdesktop può essere considerato un aggiornamento particolarmente ben voluto da tutti coloro che usano ancora il ramo Xfce 4.10. Se guardiamo la roadmap del progetto (**http://bit.ly/1xOkRZW**), scopriremo che la versione 4.12 è stata progettata a marzo 2013, ma non ancora rilasciata. Allo stato attuale, qualsiasi aggiornamento è benvoluto, sia perché dimostra che il progetto è attivo, sia perché corregge alcuni bug.

> **"Un aggiornamento benvoluto da chi usa Xfce 4.10"**



› **Xfdesktop è ora decisamente molto più ordinato anche semplicemente dal punto di vista del codice**

Programma di benvenuto su GNU

GNU Hello

Versione: 2.10 **Web:** http://ftpmirror.gnu.org/hello

La piccola applicazione **GNU Hello** viene utilizzata come riferimento per la documentazione, la struttura e il rilascio dei software sotto GNU. In pratica, serve per educare tutti i neofiti del mondo GNU's Not Unix a lavorare con i programmi in modo più ordinato ed essenziale. L'applicazione, a tal proposito, si chiama Hello in omaggio all'iconica frase "Hello, world!", utilizzata dalla maggior parte degli sviluppatori per provare i propri software. Nel caso non lo sappiate, il famoso "Ciao, mondo!" è comparso per la prima volta in una nota interna dei Bell Laboratories, in merito a un programma scritto in C da Brian Kernighan. Alla base di questo programma c'è la volontà da parte degli sviluppatori di dare maggior consapevolezza degli standard GNU. L'utility GNU Hello, peraltro, fa parte di tutte le distro Linux. Basta scrivere **hello** e si ottiene la risposta "Hello, world!" tradotto nella lingua locale del sistema. Scrivendo **hello -g <testo>**, è possibile sostituire la frase predefinita con quella che preferite. Infine, potete utilizzare **hello -t** per ottenere un saluto predefinito, ma senza il punto esclamativo. Invece di utilizzare lo GNU Hello della vostra distro, vi consigliamo di scaricarlo dal sito ufficiale e compilarlo da sorgente. Peraltro, per chi non ha mai compilato un software Linux prima, questo è un modo perfetto per cominciare. GNU Hello ha un design molto pulito ed essenziale. Utilizza **Gettext** e alcuni componenti essenziali **dev.** Tra le altre cose, è anche bizzarro vedere come il comando **./configure** di GNU Hello richieda più tempo rispetto a **make**. La nuova versione 2.10 contiene alcune traduzioni aggiornate, una pagina **man** meglio distribuita e varie migliorie a sostegno del pacchetto **Autotools**. Tra gli altri miglioramenti abbiamo notato che la funzione **fprintf** (stderr) è adesso sostituita da **error()**.

> **"GNU Hello vi aiuta a muovervi secondo gli standard GNU"**

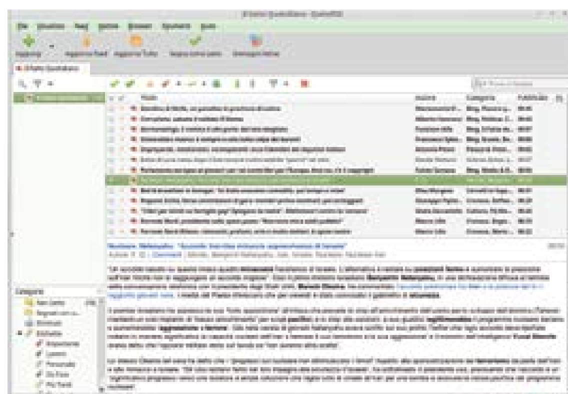


› **GNU Hello può vantare un'ottima documentazione che troviamo sul sito ufficiale www.gnu.org/software/hello**

Lettore RSS

QuiteRSS

Versione: 0.17.6 **Web:** http://quiterss.org



› Per qualsiasi azione si voglia compiere, QuiteRSS ha il pulsante giusto nella barra degli strumenti

Anziché navigare tra molteplici siti di quotidiani e notiziari, è molto meglio utilizzare un programma che aggreghi tutte le fonti e riporti gli argomenti più interessanti. Dal 1995, RSS è il sistema più comodo per leggere notizie e articoli. Il suo sviluppo è partito da *Rich Site Summary* fino a divenire l'odierno *Really Symple Syndication*. Molti siti Web, per non dire quasi tutti, fanno uso di RSS e sono progettati come pagine XML personalizzate su determinati URL. Un feed RSS può essere visualizzato sia su un browser, sia su client appositamente dedicati. Di questi ce ne sono davvero molti in Rete, ma **QuiteRSS** è uno dei migliori. L'applicazione è ricca di funzionalità e ha il pregio di mettere tutti gli strumenti che servono subito sotto mano. Il riquadro in alto a sinistra mostra l'elenco dei feed; per aggiungere una nuova sorgente, invece, è possibile utilizzare la barra degli strumenti superiore. Subito sotto a essa troviamo l'albero delle etichette colorate che consentono di contrassegnare i feed e facilitarne il recupero. La parte principale dell'interfaccia fa somigliare QuiteRSS a un client di posta elettronica. Il suo ruolo è visualizzare l'elenco delle notizie e mostrarne il contenuto. Il riquadro inferiore, infine, presenta i contenuti all'interno di un browser Web. QuiteRSS, dal punto di vista grafico, può sfruttare una GUI estremamente ben progettata che supporta modalità di visualizzazione classica e di lettura. In quest'ultimo caso sarà possibile leggere senza alcuna distrazione, nascondendo le barre degli strumenti e le finestre non necessarie. In più, si può sfruttare la funzione di ricerca avanzata, diversi metodi di selezione e una varietà di controlli sulle notifiche da ricevere. La nuova versione di QuiteRSS corregge una serie di bug e aggiunge funzionalità minori, nonché miglioramenti al design. Il sito Web contiene tutte le informazioni per l'installazione, ma anche diversi pacchetti precompilati da usare con il vostro sistema.

> “QuiteRSS è una delle migliori applicazioni per la lettura dei feed RSS”

Grabber CD audio

Audex

Versione: 0.79 **Web:** http://bit.ly/AudexApp



› Per aggiungere più codec oltre al WAV è sufficiente andare in *Configura Audex* e poi in *Profili*

Scuotete la polvere dai vostri CD musicali e convertiteli in file musicali. Naturalmente, questa idea poteva essere rivoluzionaria fino a 10 anni fa. Adesso, i CD vengono per lo più usati come oggetti d'arredamento o come cimeli di un'era passata. Ciò nonostante, scommettiamo che se siete collezionisti o amanti della musica in generale avrete diversi supporti ottici sparsi per la casa con incise le più belle canzoni italiane e non. **Audex** vi permette di dare un taglio al passato o per lo meno di crearvi un archivio multimediale completamente digitalizzato. Infatti, si tratta di un'applicazione di CD-ripping proveniente dal mondo KDE. Da semplice software, si è comunque evoluto tanto da supportare il nuovo desktop Plasma, seppure funzioni egregiamente anche su Unity o ambienti di terze parti. L'applicazione, una volta letto il CD, mostra la lista delle tracce presenti e cerca automaticamente il database CDDB, applicando poi il nome ai vari brani. Questa modalità di funzionamento permette anche di risolvere problemi piuttosto comuni, dividendo i titoli, scambiando artista e titolo e riempiendo gli eventuali campi mancanti. Il riquadro di sinistra, invece, mostra la copertina dell'album su cui state lavorando. Funziona bene, ma purtroppo è necessario scaricare manualmente la cover. Un altro aspetto che deve essere controllato a mano riguarda i profili di codifica. Per impostazione predefinita, infatti, Audex fornisce solo profili WAV. Ciò non toglie che sia possibile aggiungerne altri. Basta andare in **Impostazioni -> Configura Audex... -> Profili**, quindi premere il pulsante **Scansione codec**. A questo punto, Audex cercherà di rilevare i codec esistenti, creando poi dei profili dedicati. Ogni profilo ha una serie di impostazioni, dove è possibile specificare il bitrate, decidere se aggiungere o meno una copertina a un file e molto altro ancora.

> “Audex è uno strumento dedicato ai collezionisti di musica su CD”

Decoratore per desktop

XDecorations

Versione: 0.1.7 **Web:** http://bit.ly/XDecorations

I vecchi amministratori IT si ricorderanno sicuramente come Microsoft Plus! era in grado di rinnovare e abbellire le GUI degli anni '90. Da allora di acqua sotto i ponti ne è passata parecchia e al posto di Microsoft Plus! abbiamo **XDecoration** per Linux. Si tratta di un software molto divertente da utilizzare, pensato per decorare il vostro desktop con uno spirito festoso. Non abbiamo trovato, purtroppo, alcuna versione preconfezionata ma solo il sorgente. Infatti pare sia distribuito solo in questa forma. La compilazione, tuttavia, è molto semplice. Basta consultare il file d'installazione presente nella directory principale e, se necessario, installare alcune dipendenze, come **libXpm-dev**, **libXt-dev**, **libxext-dev** e **libimlib2-dev**. Per iniziare a utilizzare XDecoration, basta usare il file **xdecorations.rc** presente nella cartella del progetto. Si comincia con un tema dedicato al Natale:

```
./xdecorations -configfile ./xdecorations.rc -holiday Xmas -showtinsell
```

Oltre al tema natalizio, ce ne sono altri già inclusi in XDecoration: Autumn, Halloween e Weather. Tutti hanno il proprio set di elementi decorativi, ognuno dei quali può essere modificato a seconda delle preferenze (date un'occhiata alla sotto-cartella **Pixmap**). Per vedere poi i risultati del vostro estro creativo basta semplicemente minimizzare tutte le finestre. Se avete scelto il tema natalizio vi accoglierà uno splendido pupazzo di neve con una serie di luci e una spruzzata di fiocchi



› XDecorations rompe la monotonia degli sfondi per desktop con animazioni davvero particolari

che cadono dal cielo. Ogni tanto, poi, compare la slitta di Babbo Natale che passa da un lato all'altro del desktop. Per cambiare il tema, aprite il file **xdecorations.rc** e navigate tra le stringhe. Modificate poi a vostro piacimento con gli ambienti Autumn o Weather. Il file di configurazione contiene parecchie opzioni a cui mettere mano. Alcune sono commentate in modo piuttosto preciso. XDecoration, infine, è compatibile con la maggior parte dei windows manager.

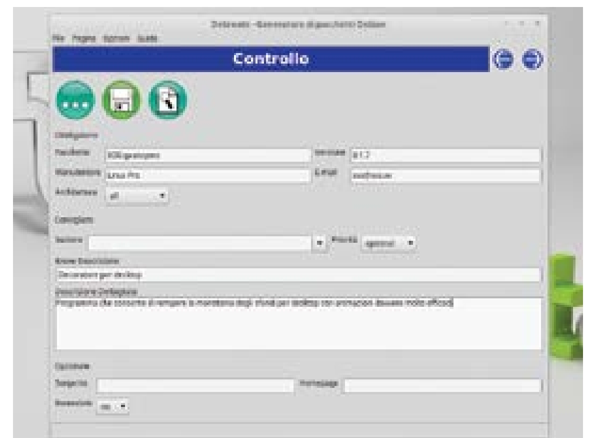
> "XDecorations è studiato per abbellire il vostro desktop"

Creatore pacchetti Deb

Debreate

Versione: 0.7.10 **Web:** http://debreate.sourceforge.net

La procedura per creare pacchetti per Debian e i suoi derivati, come Ubuntu, Mint ed ElementaryOS, è abbastanza interessante. Il motivo è presto detto: gli strumenti per eseguirla sono diversi e ognuno fornisce funzioni utili e innovative. Uno di questi è **Debreate** che offre un'interfaccia grafica pratica e intuitiva da utilizzare per creare pacchetti Deb. Chi ha qualche anno sulle spalle e un po' di esperienza nel settore troverà alcune somiglianze con Packin, un'applicazione basata su Java da cui si ispira. Debreate, tuttavia, è molto più intuitivo e consente di sfruttare una documentazione di alto livello. Anche l'installazione è molto semplice. Il sito Web del progetto offre una serie di file, tra cui quello che abbiamo provato noi: **debreate_0.7.10_all.deb**, una versione universale capace di funzionare su qualsiasi sistema basato su Debian. Al primo avvio, il programma ci informa sulle sue impostazioni, quindi mostra una schermata di benvenuto che riporta un link a un video tutorial, peraltro visibile anche sulla pagina principale del sito. Il processo di creazione di un pacchetto Deb prevede il completamento di una procedura che si snoda in diversi passaggi. In primo luogo si compila il pacchetto con i metadati, tra cui dobbiamo riempire alcuni campi obbligatori che si affiancano a quelli raccomandati e facoltativi. In seconda battuta viene mostrata la schermata delle dipendenze, dove si specificano



› Debreate permette di creare pacchetti Deb in modo semplice e intuitivo tramite una procedura passo a passo

gli altri pacchetti necessari alla corretta installazione. Andando oltre, si devono selezionare i file da includere. Qui sarà poi possibile cambiare il percorso d'installazione o applicare determinate impostazioni alle varie cartelle. La schermata successiva riguarda gli script pre e post installazione. Infine, gli ultimi passaggi prendono in considerazione il Changelog, il Copyright e il menu di avvio.

> "Un'ottima interfaccia grafica per creare pacchetti Deb"

Giochi Applicazioni per il tempo libero

Sparatutto in prima persona

Chocolate Doom

Versione: 2.1.0 **Web:** www.chocolate-doom.org

Sarete felici di sapere che in realtà questo gioco non comprende solo *Doom*, ma anche *Heretic*, *Hexen e Strife*, vale a dire tutti i più leggendari sparatutto in prima persona dei primi anni '90. Doom, tuttavia, è il principale titolo che merita di essere analizzato più da vicino. Abbiamo già visto diversi progetti che si proponevano di riportare in vita il mitico sparatutto, ma la versione Chocolate offre alcuni spunti in più. Prima di tutto si nota subito un forte spirito di conservazione per mantenere Doom in chiave minimalista, così da permettere ai giocatori di personalizzare il gioco con diversi add-on e migliorie. Un altro aspetto positivo sta nell'adattabilità del progetto. Infatti, ogni volta che il motore Open Source di Doom rimuove la compatibilità con queste mod, Chocolate le mantiene con cura. **Chocolate Doom** è rilasciato interamente sotto licenza GNU GPLv2 ed è un gioco Open in piena regola. Tuttavia, per iniziare, è necessario un file **WAD Doom** in cui siano contenuti i dati di gioco. Inoltre è disponibile per svariate piattaforme come Ubuntu, Mint, OpenSUSE, Rosa e altre distro. Il titolo gestisce tutte le versioni di Doom, tra cui The *Ultimate Doom*, *Doom 3* e *Final Doom*, nonché *Chex Quest* e *HACX*. Heretic, Hexen e Strife sono supportati anche da Chocolate Heretic, Chocolate Strife e così via. Se si dispone di più titoli (IWADS) è possibile passare dall'uno all'altro tramite il comando:

```
chocolate-doom -iwad /percorso/per/doom.wad
```

Il gioco inizia in modalità a schermo intero e ci porta indietro nel tempo come nessun altro titolo sa fare.

> "Il gioco inizia a schermo intero e ci porta indietro nel tempo"



› Preparatevi a sfoderare cannoni al plasma e armi di ogni tipo. Doom è tornato più in forma che mai

Arcade

VoltAir

Versione: 1.0.5 **Web:** http://bit.ly/VoltAir

VoltAir è un divertente e colorato gioco, dove il nostro ruolo è quello di controllare un robot a forma di pallina elettrica in fuga su un mondo sconosciuto. Durante il gameplay, il vostro robot raccoglierà diverse altre palline e guadagnerà poteri che lo aiuteranno a fuggire attraverso un portale dimensionale posto alla fine di ogni livello. Gli utenti possono giocare sia in single player sia in multiplayer locale. Il gioco è accompagnato da una piacevole colonna sonora di sottofondo e vanta una grafica a dir poco professionale. VoltAir, tuttavia, non è un altro dei tanti giochi Open Source alimentati da una comunità di fan, ma un progetto di Fun Propulsion labs, una costola di Google. Il titolo è poi rilasciato sotto licenza Apache Public. VoltAir è un cross-platform scritto in C++ che fa uso di librerie QtQuick/QML per l'interfaccia. Le piattaforme di destinazione comprendono Linux, Mac OS X e Android. Da questo punto di vista, tuttavia, vi consigliamo di provarlo su dispositivo mobile, l'unico capace di enfatizzare il gameplay con la versatilità che solo i comandi touch sanno dare. Nonostante la semplicità del gioco in sé, farlo girare su Linux non è poi così tanto facile. VoltAir, tuttavia, è disponibile anche su Google Play e può essere installato con un click su smartphone e tablet Android. Su Linux, invece, è necessario utilizzare Archon. Per compilare i sorgenti è prima d'obbligo studiare la guida ufficiale, quindi montare **LiquidFun**, una libreria fisica C++ 2D basata su Box2D che include simulazioni di particelle ad alte prestazioni. Solo così sarete in grado di compilare il gioco in QtCreator. Attualmente, infatti, non esistono pacchetti precompilati, se non per Rosa 4 e OpenMandriva Cooker.

> "VoltAir integra un simulatore di particelle ad alte prestazioni"

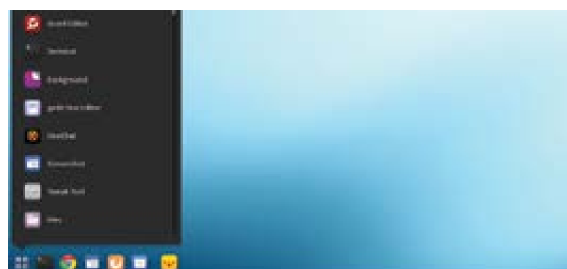


› VoltAir ha una grafica davvero ben curata e che lascia trasparire il suo sviluppo professionale

Ambiente desktop

Budgie Desktop

Versione: 0.8 **Web:** http://bit.ly/BudgieDesktop



› Budgie si distingue per i suoi componenti essenziali e decisamente minimalisti

Se siete stanchi di KDE e non potete digerire Gnome, né tantomeno riuscite a sostenere lo sviluppo oramai parecchio lento di Xfce, allora è il momento di dare un'occhiata a **Budgie**. Si tratta del fiore all'occhiello dell'ambiente desktop di EvolveOS che a sua volta è stato costruito da zero da una fork di Pisi. Budgie è disponibile in pacchetti pronti per l'installazione su Ubuntu, OpenSUSE, Fedora e Arch Linux. Gli sviluppatori di Budgie affermano che si tratta di una fork di Gnome Shell. Il codice è scritto in C e guardandolo da vicino non si può fare a meno di notare l'ordine con cui è stato steso. Per quanto riguarda il concetto di funzionamento, possiamo accomunarlo a Yorba ed ElementaryOS, ma l'esperienza per l'utente è diversa. Dopo la serie di sfondi predefiniti, è possibile notare fin da subito un pannello inferiore molto simile a quello di ChromeOS. Questo è popolato dal menu principale, l'elenco finestre, la barra delle applicazioni, il controllo volume e altri applet opzionali. Ognuno di questi può puoi essere spostato o rimosso. Basta semplicemente mettere mano alla relativa finestra delle preferenze. Le finestre sono gestite dal Budgie Windows Manager (BWM), una fork di Mutter con animazioni fluide e controllo per lo

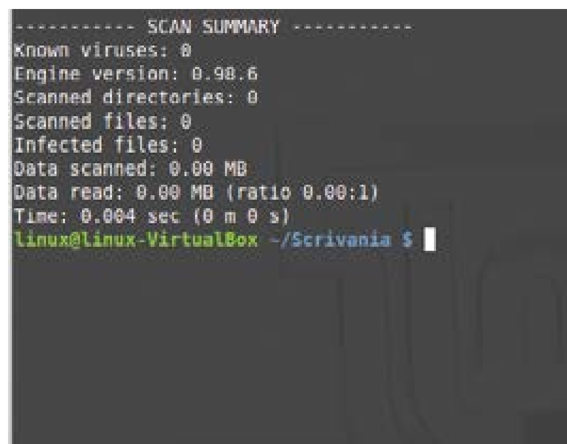
> "Se siete stanchi dei tradizionali desktop, provate Budgie"

sfondo del desktop. Molti altri componenti, come il gestore di sessione o per le notifiche, sono attualmente in fase di sviluppo e servono come front-end per le rispettive parti consolidate da Gnome. La nuova versione di Budgie ha portato con sé parecchi miglioramenti. Il menu utilizza un layout compatto e può essere configurato sia per mostrare una struttura semplice, sia avanzata con tanto di categorie. La barra inferiore, adesso, può essere nascosta automaticamente e in più supporta applicazioni di pinning, tematizzazione dinamica e molto altro ancora.

Antivirus

ClamAV

Versione: 0.98.6 **Web:** www.clamav.net



› Se state utilizzando Windows in dual boot, avere ClamAV è sempre un'ottima idea

ClamAV è una soluzione multipiattaforma per il rilevamento di virus e malware su piattaforma Linux. Fornisce un livello relativamente alto di funzionalità, anche se non tante quanto le controparti commerciali. ClamAV, infatti, è solo in grado di rilevare i programmi maligni e spostare i file infetti in una cartella di quarantena. La mancanza principale, però, è che non può eliminarli. Sebbene funzioni in modalità cross-platform, il suo utilizzo è concentrato soprattutto per rilevare virus su partizioni Windows. Viene quindi sfruttato soprattutto su NTFS o FAT in condivisione SMB. Questo programma è disponibile per quasi tutte le distro Linux. Basta installarlo tramite il vostro gestore pacchetti. Il modo più semplice e comodo per iniziare a utilizzarlo è l'esecuzione del comando **clamscam**:

```
clamscan -r /percorso/per/cartella
```

Dopo un po' di tempo (dipende dal numero di file e dalle dimensioni della cartella presa in esame) avrete un riepilogo con le informazioni sui virus rilevati, la versione del motore di scansione e le statistiche. Un altro nucleo eseguibile è il Bytecode Compiler (clambc) che viene utilizzato per compilare firme in Bytecode per l'elaborazione di file aggiuntivi. Le Bytecode, infatti, sono un tipo di firma specializzato di ClamAV. Queste sono in grado di eseguire ulteriori elaborazioni dei file

> "L'ultima versione di ClamAV porta con sé notevoli migliorie"

digitalizzati, consentendo anche una rilevazione più marcata. L'ultima versione di Clam porta con sé un notevole miglioramento per il rilevamento dei malware, includendo le proprietà di analisi dei file e utilizzando strutture API e Bytecode. Inoltre, adesso è in grado di supportare il formato XDP (*XML Data Package*), riuscendo anche a decodificare e scansionare i file PDF all'interno degli XDP. Infine, il nuovo ClamAV è capace di contenere eventuali crash dovuti all'azione di certi virus. LXP

Tutorial

I nostri esperti offrono i loro consigli di programmazione e di amministrazione del sistema

LA GUIDA DI RIFERIMENTO

Esiste sempre qualcosa di nuovo da imparare in campo informatico, soprattutto in un mondo dinamico come quello di Linux e dell'Open Source. Ogni numero di Linux Pro presenta una sezione dedicata a tutorial realizzati da esperti in moltissimi settori: programmazione, sicurezza, amministrazione di sistema, networking. Troverete informazioni utili sia che siate dei veterani di Linux sia degli utenti alle prime armi. Studieremo con cura anche le applicazioni più diffuse sia in ambito lavorativo che desktop. Il nostro scopo è quello di fornire in ogni numero il giusto mix di argomenti, ma se avete suggerimenti su temi particolari che vorreste vedere trattati, scriveteci via e-mail all'indirizzo tutorial@linuxpro.it

COME RAPPRESENTIAMO LE LINEE DI CODICE

Si presenta spesso la necessità di riportare le linee di codice di un programma. Per favorirne la lettura evidenzieremo le singole linee in questo modo:

```
begin
  mniWordWrap.Checked := not
end
```

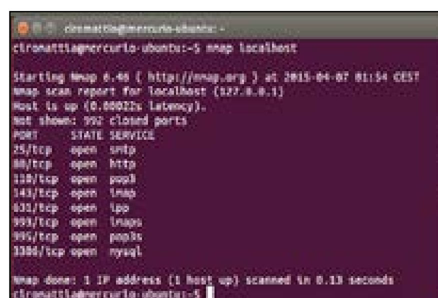
Quando una riga di codice supera la lunghezza della colonna la riporteremo su più righe utilizzando la notazione seguente:

```
printf("Vi preghiamo di inserire
una password.");
```

TUTORIAL







ACCADEMIA DEL CODICE

Controllare le intrusioni

Lo staff di Linux Pro vi mostrerà come creare una porta di servizio sicura e poi esporla a Internet per scoprire se qualcuno vi sta guardando



FACILE

Un semplice fatto: quando una porta di servizio è disponibile in Internet, qualcuno la guarda. Vi mostreremo come verificarlo, dimostrandovi quanto sia necessaria un po' di sicurezza sulla propria rete, gestendo i firewall e installando con regolarità gli aggiornamenti. Vi mostreremo anche come rendere sicura una porta del vostro computer, configurare un router domestico per fare il forward di una porta su Internet e quindi controllare i log per vedere gli IP che hanno tentato di accedervi. Il diagramma della prossima pagina mostra un modem/gateway residenziale (RG), che offre accesso alla rete a tre host, visibili sulla destra. L'RG ha un hub/switch interno e supporto Wi-Fi per offrire la connessione LAN (*Local Area Network*). Usa un DHCP (*Dynamic Host Control Protocol*) per distribuire gli indirizzi IP ai vari dispositivi che si connettono al RG dalla LAN. DHCP è il protocollo responsabile della distribuzione e gestione degli indirizzi IP assegnati agli ospiti di una LAN. Il range di indirizzi che utilizza il DHCP è valido soltanto per i tre dispositivi privati. I loro indirizzi IP sono validi soltanto nel lato LAN del RG e non è permesso propagarli in Internet. Nella parte sinistra troviamo la WAN (*Wide Area Network*), la rete di telecomunicazioni che fornisce accesso a Internet. Affinché l'RG funzioni nella WAN deve avere un indirizzo IP pubblico che gli viene assegnato dall'*Internet Service Provider* (ISP). Un range di IP pubblici viene assegnato a ogni ISP che è responsabile della loro gestione. L'ISP ha il suo DHCP che distribuisce e gestisce gli IP pubblici da assegnare ai dispositivi dei clienti. NAT (*Network Address Translation*) è il meccanismo utilizzato per inviare il traffico degli IP della LAN su Internet. Le richieste di accesso a Internet provenienti dai vari IP privati all'interno della LAN vengono tradotte tramite NAT all'IP pubblico assegnato all'RG per essere inviate sulla Rete. Una tabella di stato all'interno dell'RG tiene traccia di quali richieste sono state inviate, così le risposte ricevute da Internet possono venire tradotte dall'IP pubblico all'IP privato e inviate al dispositivo corretto all'interno della LAN. Per vedere l'indirizzo IP privato assegnato al vostro computer è sufficiente aprire un terminale e digitare il seguente comando:

```
sudo ifconfig
```

In questo articolo vedremo come esporre in Internet la porta standard SSH (22). SSH è un protocollo di rete che offre comunicazioni sicure tra due computer su reti insicure. Molti amministratori di rete utilizzando SSH per fare login da remoto sui computer. Per assicurarsi che la suite del protocollo SSH sia installata su un computer Unix, basta lanciare questo comando:

```
whereis ssh
```

Se questo comando ritorna una directory come **/usr/bin/ssh**, il servizio è disponibile. Se il comando non ritorna il percorso dell'SSH potete utilizzare il vostro gestore di pacchetti per installare il programma. Su sistemi Linux derivati da RedHat/Fedora vi basterà digitare:

```
sudo yum install ssh
```

Per i sistemi Debian based invece:

```
sudo apt-get install ssh
```

Dato che la nostra porta SSH sarà disponibile su Internet, deve essere sicura per evitare che il sistema venga compromesso durante il nostro test. Per potersi connettere al servizio SSH, sono necessari un nome utente e una password. Il nome utente utilizzato di solito per fare i tentativi è **root**. Con un editor di testo aprite il file di configurazione **/etc/ssh/sshd_config** e cercate la riga contenente il testo **PermitRootLogin**. Assicuratevi che quella riga contenga soltanto: **PermitRootLogin no**. Salvate e uscite dal file. Un'ultima modifica alla configurazione da eseguire per assicurarci che la porta esposta sia sicura è quella di disabilitare tutti i tentativi di login. Create questa voce nel file **/etc/hosts.deny**: **ALL:ALL**. **TCP_Wrappers**

Tip

Potete trovare un elenco delle porte associate ai servizi nel file ***/etc/services***. Ovviamente questo vale per Unix, non è altrettanto vero per altri SO come Windows.

Un posto per gli username

Facendo un cat di **/etc/passwd** dal terminale vi mostrerà tutti gli account del vostro sistema. Alcuni username sono associati a dei servizi. I tentativi di accesso al sistema necessitano di un username e di una password. Utilizzando username comuni creati per servizi, applicazioni o database presenti nel file **/etc/passwd** è un modo per risolvere metà dell'equazione username/password. È importante per un amministratore sapere quali applicazioni sono installate sul computer per assicurarsi che le password di default vengano cambiate dopo l'installazione. Un esempio di un'applicazione che contiene un username e password di default è **MariaDB**. Se quando si installa il database non si cambiano le impostazioni di default, gli utenti malintenzionati avranno tutti i componenti necessari per autenticarsi.

Log dai servizi

Il servizio SSH ha un suo log di informazioni. Il log contiene dettagli sull'avvio del servizio e dati sia sulle connessioni eseguite che su quelle fallite. I log in Linux hanno subìto diversi cambiamenti di recente. **Syslog** e **rsyslog** si occupano da sempre dei log, ma ora c'è un nuovo arrivato **journald**. Il posto in cui reperire i log dipende dal sistema in uso. Se il vostro sistema utilizza un sistema di log classico troverete un file chiamato **secure** da qualche parte in **/var/log/**. Utilizzando un editor di testo potete esaminare il file. Se il vostro sistema invece utilizza journald, dovrete utilizzare il tool **journalctl tool** per leggere i file binari dei log di sistema.

è un servizio che utilizza i file **/etc/hosts.deny** ed **/etc/hosts.allow** per restringere l'accesso a certi servizi. Molto prima dei firewall personali, **TCP_Wrappers** era la difesa principale per proteggere i servizi. ALL:ALL disabilita tutti i servizi per tutti gli utenti. Mettendo invece **sshd:ALL** si disabilitano solo i servizi SSH a tutti gli utenti. Potreste pensare che la voce **PermitRootLogin no** nel file di configurazione di SSH sia ridondante con l'impostazione di TCP_Wrappers, e avreste ragione. Purtroppo TCP_Wrappers sta cadendo in disgrazia e alcune varianti di Linux hanno scelto di non utilizzarlo più. Potete decidere di non preoccuparvi del controllo sui servizi offerto da TCP_Wrappers e non usarlo. Con SSH sicuro potete avviare il relativo demone con il seguente comando. Nei derivati di RedHat:

```
sudo service sshd start
```

In quelli Debian

```
sudo service ssh start
```

Come controllo veloce per vedere se SSH è in esecuzione, utilizzate il comando **netstat** per vedere se la porta 22 è in ascolto:

```
netstat -an | more
```

Il comando netstat con lo switch **-an** mostra le informazioni sulle connessioni di rete senza utilizzare il DNS. Se lanciate il comando senza lo switch **n**, otterrete il nome del servizio associato alla porta anziché il numero (dal file **/etc/service**). Cercate nell'output del comando **netstat -an** una riga simile a

```
tcp    0    0 0.0.0.0:22        0.0.0.0:*        LISTEN
```

oppure

```
tcp    0    0 0.0.0.0:ssh        0.0.0.0:*        LISTEN
```

Se la porta è in ascolto potete provare a connettervi utilizzando il seguente comando impostando l'IP privato dell'host Linux.

```
ssh root@<indirizzo IP privato>
```

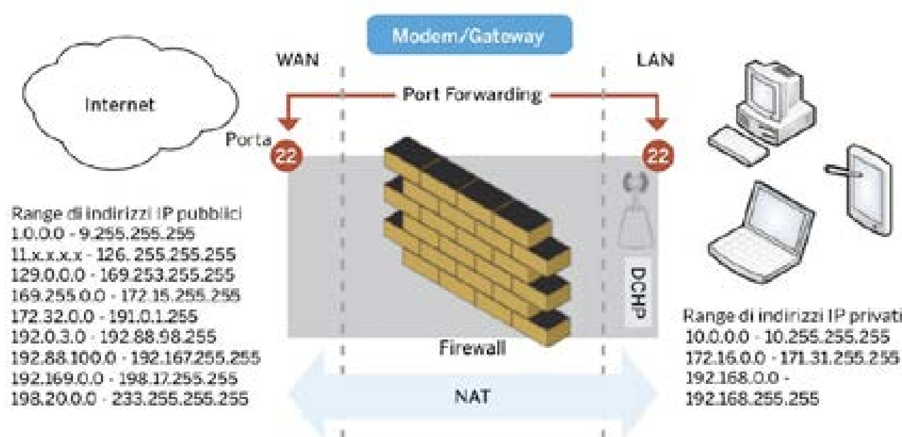
Se avete configurato correttamente SSH per inibire il login dell'utente root e/o avete impostato la sicurezza di TCP_Wrappers come detto prima, il comando non dovrebbe funzionare. Quindi possiamo assumere che sia sicuro. Ora esponiamo la porta in Internet. Lo faremo utilizzando il port forwarding, il meccanismo che utilizzano gli RG per prendere le richieste fatte a una porta dal lato WAN e inviarle a uno specifico host nel lato LAN. Come mostra il diagramma, quello che vogliamo fare è configurare il modem per inoltrare tutte le richieste SSH ricevute da Internet sulla porta 22 al nostro host. I passi da seguire per farlo dipendono dal vostro modem e dalla versione del suo firmware, quindi dovrete utilizzare la documentazione del vostro modem per sapere come fare. Probabilmente vi serviranno i permessi di root sul vostro RG per farlo. Con l'RG e il computer configurati, lasciate la porta aperta per alcune ore, quindi seguite la procedura illustrata qui sotto per vedere se ha attirato qualche attenzione. In questo esempio ci limiteremo a esaminare il file **/var/log/secure**. Con i comandi appropriati potete vedere informazioni simili da altri sistemi di log (vedi box qui sopra).

Esaminare i log

Quando viene tentato l'accesso su di una porta esposta, viene creato un record SSH nel log, contenente l'indirizzo IP del visitatore a cui è stato rifiutato l'accesso. Basta utilizzare un editor di testo per esaminare il file e vedere queste informazioni. Potete poi utilizzare un servizio come **https://www.arin.net/** inserendo l'IP del vostro quasi-visitatore nel campo **SEARCH whois** per ottenere alcune informazioni sul suo provider. Secondo la nostra esperienza troverete tentativi di connessione da tutte le parti del mondo. Non abbiamo approfondito a sufficienza i nostri esami per poter dire se si tratta di attacchi coordinati o di computer compromessi di qualche sfortunato utente domestico. Come potete vedere la nostra configurazione prevede un firewall ma in questa circostanza serve a ben poco. Come abbiamo visto l'RG inoltra le connessioni in arrivo (dalla WAN alla LAN) soltanto se sono state espressamente impostate dall'amministratore. Il firewall interviene se decidete di controllare i servizi in uscita. Se l'RG è in modalità PPPoE pass-through il firewall diventa vitale per restringere gli accessi all'host. Il semplice esercizio che vi abbiamo mostrato è servito per dimostrarvi che una porta esposta in Internet attrae dei visitatori anche se non li avete informati. Questo è uno dei motivi per cui la sicurezza è vitale per proteggere i servizi dalla Rete. C'è sempre qualcuno che vi sta guardando! LXP

Tip

Se il vostro dispositivo LAN ha un indirizzo IP pubblico, il vostro RG è configurato in modalità PPPoE pass-through. Gli indirizzi IP privati non vengono assegnati utilizzando DHCP, e gli indirizzi IP pubblici passano attraverso il RG fino al computer. Questo tipo di configurazione rende l'utilizzo di un firewall vitale.



SquirrelMail: un'introduzione

Ecco come creare il vostro personale servizio Webmail da zero, installando e configurando **SquirrelMail**



INTERMEDIO

Che sia per maggiore sicurezza o semplicemente per maggiore controllo, potreste voler gestire da voi il vostro server mail. Ecco come installare SquirrelMail così da inviare e ricevere email dal vostro Web server con un comune browser, a prescindere da dove vi trovate. Se avete una macchina LAMP preimpostata potete saltare direttamente alla sezione "Il server email". In caso contrario, ecco come impostare Apache, PHP e MySQL in un minuto o due:

```
sudo apt-get install apache2
sudo apt-get install mysql-server libapache2-mod-auth-mysql php5-mysql
sudo apt-get install php5 libapache2-mod-php5 php5-mcrypt
```

Il server email

Per cominciare a gestire le vostre email dovete installare un server email. Ce ne sono diversi a disposizione, incluso **Exim**, ma qui verrà usato **Postfix**. Per installarlo lanciate i comandi seguenti:

```
root@mercurio-ubuntu:~# apt-get install postfix postfix-mysql dovecot-core dovecot-imapd dovecot-lmtpd dovecot-mysql dovecot-pop3d
```

Durante l'installazione potete crearvi un certificato autofirmato oppure scegliere di non avere SSL. Quando arrivate all'hostname, potete lasciare il default (localhost). Se pensate di usare un nome di dominio aggiungetelo sotto a "System Mail Name", perciò se un indirizzo nell'host locale è nome@example.org il valore corretto sarà **example.org**. Una volta impostato il vostro server email, il passo successivo è installare il pacchetto per la Webmail. SquirrelMail è leggero, ha una lunga storia ed è spesso usato da grosse aziende. Per installarlo lanciate il comando seguente come root:

```
# apt-get install squirrelmail squirrelmail-locales
```

Per configurare SquirrelMail potete usare semplicemente il comando **squirrelmail-configure**. Prima, comunque, dovete fare qualcos'altro per far funzionare correttamente le cose. Innanzitutto un link simbolico:

```
# ln -s /etc/squirrelmail/apache.conf /etc/apache2/conf-enabled/squirrelmail.conf
```

(Notate che qualche mese fa per fare la stessa operazione avreste dovuto usare un comando leggermente diverso; i nomi delle cartelle e dei file vengono però aggiornati con il tempo, quindi la directory che prima si chiamava **conf.d** ora è **conf-enabled**). Proseguendo, lanciate un **service apache2 restart**. Ora dovrete poter aprire il vostro browser, scrivere **http://localhost/squirrelmail/src/login.php** e vedere la pagina di login. Anche se non potete ancora accedere effettivamente, se vedete questa pagina l'installazione è andata a buon fine. La pagina da aprire ora è **http://localhost/squirrelmail/src/configtest.php**. È possibile che vediate un errore IMAP:

```
ERROR: Error connecting to IMAP server "localhost:143".
Server error: (111) Connection refused
```

Non vi spaventate, questo errore è facilmente risolvibile. Per farlo lanciate per prima cosa:

```
apt-get install nmap
```

Quindi lanciate il seguente comando come root per scansionare le porte:

```
nmap localhost
```

Vi verrà ritornato un elenco di porte e potrete vedere se qualcuna manca. Per esempio, se non vedete la 143 o la 25 dovreste pensare a come risolvere i problemi dei vostri server IMAP o SMTP. Usando SquirrelMail, **Dovecot** è solitamente il problema principale, potete ricaricarlo con

```
service dovecot reload
```

Un altro modo facile per identificare i problemi di IMAP è il comando

```
sudo netstat -a | fgrep imap
```

Tale comando ritorna una riga contenente LISTEN. Se è così, sapete che il server IMAP è correttamente in funzione. Un'altra alternativa è lanciare il comando **netstat -nl4**, quindi per controllare se riuscite a collegarvi sulla porta 143, lanciate il comando

```
telnet localhost 143
```

(Se non avete telnet installatelo con **apt-get install telnetd**). Se telnet ha successo, vedrete il messaggio 'Connected to Localhost' e potrete andare oltre. Ricaricate Dovecot con il comando usato poc'anzi. A questo punto SquirrelMail dovrebbe essere utilizzabile. Se aprite ora la pagina **http://localhost/squirrelmail/src/configtest.php** nel vostro browser dovreste vedere una riga in fondo che recita 'Congratulations, your SquirrelMail setup looks fine to me!'

Impostare SquirrelMail

Provate ad accedere a **http://localhost/squirrelmail/src/login.php** con il vostro username e password Ubuntu. Se funziona dovreste vedere la pagina webmail.php e molto probabilmente un errore riguardante la INBOX. Tuttavia gli altri collegamenti (INBOX.Drafts, INBOX.Sent e INBOX.Trash) dovrebbero funzionare correttamente. Se qualcosa va storto

Tip

Se il vostro ISP non permette al vostro mail server di spedire mail sulla porta 25 potete sempre chiedergli di modificare l'impostazione oppure inviare le mail in uscita tramite un provider come Gmail.

Usare nmap

Usare **nmap** vi aiuterà a scansionare le porte nel vostro localhost. Potete usarlo per scansionare qualsiasi sito Web dal momento che è uno strumento eccezionale per il controllo della sicurezza di rete. Se usate solamente il comando **nmap** seguito dall'host destinazione (localhost, ip, url) scansionerà 1.000 porte, un ampio spettro per il vostro esercizio.

```
# nmap localhost
Starting Nmap 6.46 ( http://nmap.org ) at 2014-11-20 13:35 PST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000068s latency).
Not shown: 989 closed ports
```

PORT	STATE	SERVICE
22/tcp	open	ssh
23/tcp	open	telnet
25/tcp	open	smtp
80/tcp	open	http
110/tcp	open	pop3
143/tcp	open	imap
631/tcp	open	ipp
783/tcp	open	spamassassin
993/tcp	open	imaps
995/tcp	open	pop3s
3306/tcp	open	mysql

A parte l'esercizio, nmap è un ottimo strumento per la scansione delle porte che può essere usato per capire i particolari degli host nel Web, come porte aperte, tecnologia in uso e versione, sistema operativo e traceroute.
Le tre righe seguenti possono essere usate per scansionare porte individuando la versione. Il secondo comando scansionerà solamente la porta 4000 mentre l'ultimo scansionerà un range di porte.

```
sudo nmap -sV <indirizzo/ip>
sudo nmap -sV -p 4000 <indirizzo/ip>
sudo nmap -sV -p 4500-5000 <indirizzo/ip>
```

Per un elenco di opzioni di nmap visitate **http://linuxcommand.org/man_pages/nmap1.html**.

troverete gli errori nel file **/var/log/mail.log**. Un errore comune è che l'utente non ha il file INBOX nella posizione corretta. Il log di errore evidenzierà che non è stato possibile creare la inbox:

```
Apr  5 19:56:41 mercurio-ubuntu dovecot: imap(ciromattia): Error: Failed to autocreate mailbox INBOX: Internal error occurred.
```

Dovrete creare il file INBOX in **/var/mail/nomeutente**. Il file dove potete modificare quest'impostazione è **/etc/dovecot/conf.d/10-mail.conf** e **mail_location** specifica il locale per la INBOX. La proprietà può essere di quell'utente ('username' per default) e i permessi 777. Dev'essere un file vuoto e non una directory. Potreste dover fare un paio di altre modifiche per far funzionare SquirrelMail. La posizione della mail è critica per indicare dove sono la vostra mailbox e la inbox. Potete capire dov'è posizionata la vostra mailbox con il comando:

```
# grep -r "mail_location" /etc/dovecot
```

In questo caso punta a **/etc/dovecot/conf.d/10-mail.conf**. I due file principali che dovrete modificare sono **/etc/postfix/main.cf** e **/etc/dovecot/conf.d/10-mail.conf**. A questo punto dovrebbe funzionare tutto correttamente. Se inviate un'email tuttavia è poco probabile che arrivi a destinazione. Il server basilare è impostato, ma per far funzionare correttamente le email dovete abilitare il port forwarding della porta 25 del router sul vostro IP locale. Se non viene fatto il forwarding dell'IP potrete inviare mail ma non riceverle. È semplice: entrate nel router e fate il forward dell'IP locale per accettare l'indirizzo. SquirrelMail ora dovrebbe funzionare. Potete impostare il vostro nome e indirizzo email in **Options** -> **Personal informations**. Aprite un terminale e passate all'utente root; quindi aggiungete un nuovo utente e impostatene la password:

```
useradd test
passwd test
```

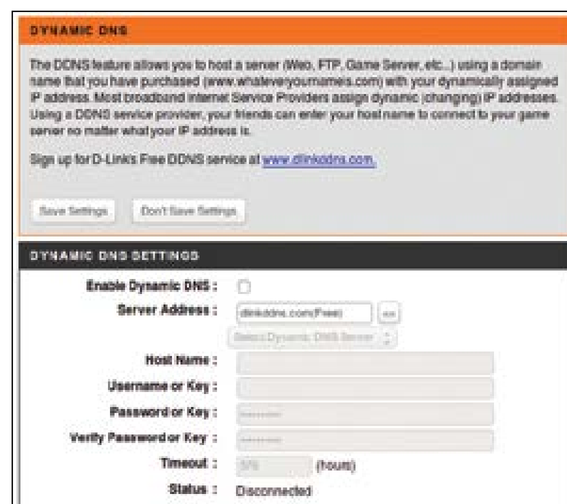
Create quindi una nuova cartella per l'utente all'interno della directory home e aggiungete anche una directory per la mail. Le istruzioni sono qui sotto e includono l'impostazione della proprietà e dei permessi.

```
mkdir /home/test
mkdir /home/test/mail
chown test:test mail
chmod -R 777 mail
```

Non vi preoccupate per INBOX.Drafts, INBOX.Sent e INBOX.Trash visti prima: verranno creati in automatico. A questo punto dovreste riuscire a inviare mail da un utente all'altro.

Inviare a server esterni

Spedire email a un altro utente con il vostro dominio è una cosa; spedire mail oltre la vostra rete locale è un'altra. L'ostacolo più grosso è normalmente la cara vecchia porta 25. Potrebbe essere bloccata dal vostro ISP, nel qual caso quest'ultimo potrebbe avere anche la soluzione. In alcuni casi potrebbe addirittura sbloccarvela. Potreste anche usare Telnet per vedere se è bloccata o meno. In alcuni rari casi il vostro ISP potrebbe essere in qualche blacklist, cosa che potrebbe rendervi ben più problematica l'impostazione del vostro server email. A questo punto avete due opzioni tra cui scegliere: la prima è spedire le email direttamente dal vostro server email, l'altra è usare un relay come Gmail per l'invio. La seconda opzione è ottima se il vostro IP è in blacklist o se il vostro ISP blocca la porta 25. Per rendervi la vita più semplice avete due modelli di file per **main.cf** (uno con relay e uno senza) e funzioneranno qualsiasi sia la vostra scelta. Con entrambi i file dovrete modificare i parametri **myhostname** e **myorigin** con il vostro dominio. Gli esempi usano example.org come dominio, quindi controllate e sostituitelo con il valore corretto. In aggiunta dovrete aggiungere l'IP della vostra rete al parametro **mynetworks**. Il segnaposto è 192.168.0.109 ed è scritto alla fine della riga con lo spazio iniziale. Se la porta 25 non è bloccata dal vostro ISP, potete inviare e ricevere email senza un relay. Se decidete di percorrere tale strada potete »



› **Alcuni router offrono un servizio DNS gratuito così da poter usare il vostro IP casalingo come hosting o come server mail**

Tip

MX Toolbox è una risorsa fantastica per decifrare lunghi header email e controllare le spam list. Lo trovate all'indirizzo **http://mxtoolbox.com/EmailHeaders.aspx**.

usare il codice sottostante per il file di configurazione **/etc/postfix/main.cf** (sono stati rimossi tutti i commenti):

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no
append_dot_mydomain = no
readme_directory = no
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = example.org
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = example.org
mydestination = example.org, mercurio-ubuntu, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.109
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

Se non potete inviare email tranne che ad altre email nel vostro nuovo server mail, potete usare un relay per l'SMTP. Per farlo dovrete fare le tre modifiche seguenti al file **/etc/postfix/main.cf** su nuove righe:

```
relayhost = [smtp.gmail.com]:587
smtp_use_tls=yes
```

e

```
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
```

Se siete nuovi a questo genere di cose, date un'occhiata a **/usr/share/postfix/main.cf.dist** per una versione completa e commentata del codice necessario (già che ci siete, guardate anche **/usr/share/doc/postfix/TLS_README.gz** nel pacchetto **postfix-doc** per informazioni sull'abilitazione di SSL nel client SMTP). Fatte le dovute modifiche al file **main.cf** e ai file **/etc/postfix/sasl_passwd**, lanciate il comando:

```
$ sudo postmap /etc/postfix/sasl_passwd
```

Questo comando serve per far sapere a Postfix dove trovare il file con lo username e password di Gmail. Il codice nel file **sasl_passwd** sarà analogo alla riga sottostante tranne il fatto che dovrete usare una password reale del vostro account Gmail.

```
[smtp.gmail.com]:587 nomeutente@gmail.com:lamiapassword
```

Ora dovrete solamente ricaricare Postfix e siete pronti!

```
sudo service postfix reload
```

A questo punto, quando spedite un'email, il vostro server email farà relay e si autenticherà sul vostro account Gmail e il messaggio verrà smistato al destinatario desiderato. (Notate comunque che i messaggi in entrata non saranno rispediti con Gmail o altri servizi impostati per il relay: le email spedite alla vostra inbox ci finiranno direttamente dal mittente). Se spedite le email tramite un relay, gli header delle stesse conterranno informazioni sia del mittente originale che dell'account Gmail. Un sito per identificare lunghi header è **http://mxtoolbox.com/EmailHeaders.aspx**. Il destinatario vedrà nel campo **Return-Path** che il relay è un account Gmail, ma non è un modo per nascondere il mittente: se analizzate gli header con MX Toolbox o manualmente, riuscirete tranquillamente a tracciare il percorso e capire che l'email è partita (in questo caso) da ciromattiaòexample.org; l'header **X-Google-Original-From** indica che Google l'ha ricevuta da qui. I campi **Received**

› Lanciare *nmap localhost* può fornirvi informazioni vitali sulle porte che state usando. Controllate la 25 e la 143

```
ciromattia@mercurio-ubuntu: ~
ciromattia@mercurio-ubuntu:~$ nmap localhost

Starting Nmap 6.46 ( http://nmap.org ) at 2015-04-07 01:54 CEST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00022s latency).
Not shown: 992 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
80/tcp    open  http
110/tcp   open  pop3
143/tcp   open  imap
631/tcp   open  ipp
993/tcp   open  imaps
995/tcp   open  pop3s
3306/tcp  open  mysql

Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
ciromattia@mercurio-ubuntu:~$
```

Sfruttare un client di posta

Anche se la Webmail è conveniente e accessibile da qualsiasi punto, potreste voler usare il vostro client mail preferito come **Thunderbird** per accedere alla vostra Webmail sulla vostra Linux box. Per default, Thunderbird Mail imposterà il vostro account con IMAP, per cui avrete sempre una copia dei vostri messaggi vecchi e nuovi. Tuttavia potete usare POP e aggiungere un port forwarding sul vostro router per ricevere le email sulla porta 110.

Allegati ricevuti

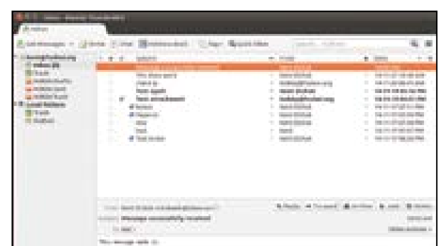
Per default un allegato sarà codificato in blocchi di testo base64 e memorizzato nella mailbox su **/var/mail/nomeutente**. Se vi inviate da soli un allegato, andate alla fine del file e leggetelo. Per arrivare alla fine del file con l'editor **Vi** scrivete **G** e date **Invio** sulla tastiera. Se volete salvare i vostri allegati potete trasferirli nella cartella **/var/lib/squirrelmail/attachments** o semplicemente scaricarli nel vostro PC. Il blocco sottostante mostra dettagli su un allegato ricevuto:

```
Content-Type: application/vnd.oasis.opendocument.text;
    name=documento.odt
Content-Disposition: attachment; filename=documento.odt
Content-Transfer-Encoding: base64
```

Se per una qualsiasi ragione volete svuotare il file inbox potete usare il comando seguente.

```
sudo truncate -s 0 /var/mail/username
```

E gli allegati inviati? Ancora, come quelli ricevuti, ognuno sarà codificato in base64 e memorizzato nel file **/home/nomeutente/INBOX.Sent**. Per vederlo da voi, spedite una mail con un file allegato, quindi aprite il file summenzionato con un editor come Vi e andate alla fine del file: troverete qui l'allegato in base64.



mostrano altresì le organizzazioni e i server che hanno gestito il messaggio in transito e includono l'IP del server email originale.

Allegati, DNS e SSH

Ora che potete spedire e ricevere correttamente le email, potreste voler potenziare il vostro SquirrelMail e offrire un modo per inviare e ricevere allegati email. È molto semplice: per cominciare, andate nella cartella **/var/lib/squirrelmail**; una volta qui, potete creare una directory per gli allegati con permessi 777 e proprietà (utente e gruppo) **www-data**.
Le istruzioni sono qui sotto:

```
# mkdir attachments
# chmod -R 777 attachments
# chown www-data:www-data attachments
```

Se state usando un Raspberry Pi, un pendrive USB, una partizione Linux o un PC con solamente Linux installato, potete impostare l'IP come statico così da non dover cambiare nulla a ogni riavvio del computer o reset del router. Ecco come fare. Il vostro nuovo servizio email è all'indirizzo locale **localhost/squirrelmail/src/login.php**. Se ci volete accedere da Internet, dovrete usare un indirizzo del tipo **http://51.68.8.248/squirrelmail/src/login.php** (naturalmente se pensate di usare il vostro IP dal vostro ISP, ci sarà tale IP al posto di 51.68.8.248). Se volete usare un nome di dominio, dovrete completare un paio di passaggi extra per far rispondere SquirrelMail da un indirizzo come **http://example.org/squirrelmail/src/login.php**. Esistono diversi servizi DNS gratuiti dove potete impostare i nameserver perché puntino al vostro IP casalingo; il vostro router potrebbe addirittura già implementarne uno. Oltre al servizio DNS gratuito, potete installare il pacchetto Linux **Bind**; quest'ultima soluzione probabilmente è un po' troppo grossa e proibitiva per un piccolo sito Web. Potete pertanto usare un servizio DNS gratuito e far puntare al vostro IP o a un VPS o server dedicato. Fatto questo, tutto si riduce a configurare il vostro router in modo che faccia forwarding della vostra Linux box: la macchina che avrà il forwarding della porta 80 sarà quella che risponderà al vostro nome di dominio. Se volete mantenere un IP statico per la vostra macchina Linux potete modificare il file **/etc/network/interfaces** e impostare un IP statico. Potete anche andare oltre entrando nel vostro router e riservando tale IP per la macchina. Il semplice codice sottostante mostra le righe due e tre originali commentate, mentre le righe seguenti sono nuove e fanno sì che l'IP resti fisso nella macchina. Con queste modifiche, avviando il sistema questo userà l'indirizzo di rete 192.168.0.109. Il gateway è l'indirizzo IP usato dal vostro router.

```
# interfaces(5) file used by ifup(8) and ifdown(8)
#auto lo
#iface lo inet loopback
auto lo eth0
iface lo inet loopback
iface eth0 inet static
        address 192.168.0.109
        netmask 255.255.255.0
        gateway 192.168.0.1
```

Per rendere effettive le modifiche lanciate il comando

```
# service networking restart
```

Ora che avete fatto tutto, potreste voler installare ils erver ssh così da potervi connettere e gestire il sistema da qualsiasi posizione vi troviate. Vi basta una riga di comando:

```
$ sudo apt-get install openssh-server
```

Ricordate di permettere il forwarding della porta 22. Seguendo tutti gli altri esempi fatti finora, l'indirizzo è 192.168.0.109. Potreste fare il forwarding di tutte le diverse porte menzionate nell'articolo. Se avete SSH attivo 24/7 potete modificare qualsiasi dei vostri file in qualsiasi momento vogliate. Per utenti remoti è praticamente impossibile immaginare una vita senza.

Gestire lo spam

Questo è tutto quello che vi serve per un server email pienamente funzionale che vi permetta di inviare e ricevere email. C'è tuttavia un'altra considerazione nel mondo attuale, e questa è lo spam. Anche se potete installare pacchetti quali **SpamAssassin** per filtrare lo spam, potete cominciare dalle cose semplici ed eliminare un sacco di spam con Postfix. Un file utile è il già visto **/usr/share/postfix/main.cf.dist**, che vi può aiutare a scegliere e conoscere vari parametri. Uno di questi è **smtpd_recipient_restrictions**, che può addirittura controllare delle spam list. Un'ottima pagina Web che aiuta moltissimo è **http://www.postfix.org/postconf.5.html**. Come vi spiegherà questa pagina, potete creare una whitelist o blacklist personalizzata e addirittura un elenco di utenti ai quali è permesso inviare email dal vostro server. Alternativamente, provate SpamAssassin come indicato sotto. LXP

Tip

Usare un IP statico e riservarlo nel vostro router può risparmiarvi molti problemi in confronto al modificare gli IP della vostra LAN. Se il vostro IP cambia, dovete modificare il port forwarding nel router e modificare i file di configurazione perché contengano l'IP corretto.

Usare SpamAssassin

SpamAssassin è un pacchetto libero per filtrare lo spam. Molte compagnie che forniscono hosting Web lo usano SpamAssassin con un server mail come *Postfix* o *Exim*. Sotto trovate un elenco di comandi utilizzabili per installare, abilitare e far funzionare SpamAssassin con Postfix. Questo tutorial è molto breve per cui utilizzerà diverse impostazioni predefinite. Se volete mettere le mani in pasta con la configurazione, le opzioni potenziali e le spiegazioni sono sufficienti per scriverci un libro.

```
sudo -s
apt-get install spamassassin spamc
groupadd spamd
useradd -g spamd spamd
mkdir /var/log/spamd
chown spamd:spamd /var/log/spamd
vi /etc/default/spamassassin
## Modificate
ENABLED=0
## in
ENABLED=1
```

Il file **/etc/default/spamassassin** è dove configurate le molte opzioni. Come potete vedere sotto, **-s /var/log/spamd/spamd.log** è stato aggiunto alla configurazione predefinita per scrivere nel file **spamd.log**.

```
OPTIONS="--create-prefs --max-children 5
--helper-home-dir -s /var/log/spamd/spamd.log"
```

Potete trovare un elenco completo di opzioni nel sito **http://spamassassin.apache.org/full/3.0.x/dist/doc/spamd.html**. Per lanciare il servizio SpamAssassin usate il comando seguente:

```
sudo service spamassassin start
```

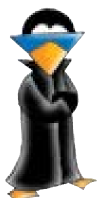
Ora dovete fare solamente una piccola modifica al file **/etc/postfix/master.cf** come mostrato sotto. Come potete vedere, la riga che inizia con **submission** non è commentata ed è stata aggiunta una nuova riga con **-o content_filter=spamassassin** con uno spazio alla sinistra. Senza lo spazio prima della riga otterrete un errore nel ricaricare Postfix:

```
submission inet n - - - - smtpd
 -o content_filter=spamassassin
# -o syslog_name=postfix/submission
```

Completate queste modifiche lanciate il comando **service postfix reload** e siete pronti!

Micro Python: un'introduzione

Come sfruttare la potenza di **Micro Python** per trasformare il piccolo ma potente microcontroller Pyboard in un mouse USB



DIFFICILE

Micro Python e il microcontroller Pyboard sono stati finanziati su Kickstarter nel tardo 2013 (**http://kck.st/17T8bW6**). L'idea era quella di rendere Python più piccolo ed efficiente a sufficienza per funzionare bene in microcontroller. Perché? Python fa un sacco di cose magnifiche ma non è per niente adatto a operazioni time-critical. Pensate solamente che Google usa Python per molte cose, ma i freni nella sua macchina a pilota automatico non sono una di queste. Riducendo il core di Python e modificando il funzionamento del compilatore, oltre ad altre cose, Micro Python tenta di essere più veloce ed efficiente di quanto potrebbe mai fare Python. Compilato in un PC reale è straordinario e potrebbe rapidamente diventare una soluzione ideale per grosse applicazioni commerciali.
In termini di usabilità e performance il microcontroller Pyboard sta da qualche parte tra un Raspberry Pi e un Arduino. Il Pi è ben più potente a confronto di un microcontroller medio, ma non è così adatto per operazioni time-critical (come i freni di una macchina). Al contrario, Arduino è piuttosto semplice, ma perfetto nel controllare input e output multipli con precisione svizzera. Con il suo veloce processore Cortex M4, un'unità per calcoli in virgola mobile a 32 bit, un generatore di numeri casuali hardware e praticamente tutte le forme di comunicazione esistenti (incluso I2C), Pyboard ha sia muscoli sia cervello.
Considerando che potete programmarlo con la stessa sintassi Python che già conoscete, Pyboard è anche davvero facile da usare. La board ha 30 GPIO (inclusi 14 input analogici a 12 bit e due output analogici), un sensore di temperatura, un clock realtime e un accelerometro a 3 assi; con tutto questo pesa solamente 3 grammi e misura solo 33 x 40 mm.
È sufficientemente piccola da essere usata per dispositivi wearable e in qualsiasi progetto vi venga in mente. Non sorprende che costi qualcosa in più di Arduino (circa 47 €), ma vale ogni centesimo.

> **Tip**
> È molto veloce e semplice dialogare con l'hardware attraverso una connessione seriale. Utilizzando **Screen** potete aprire una prompt di Python con **screen /dev/ttyAMCO**.

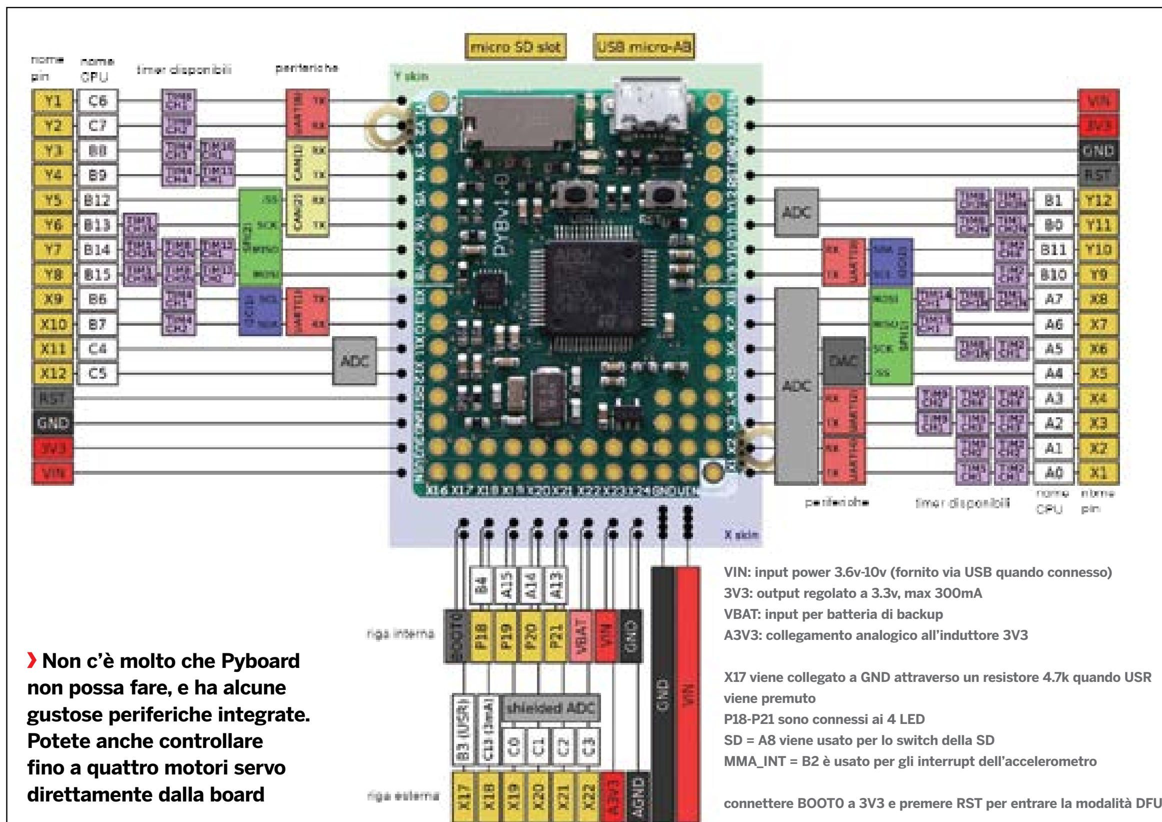
Accensione

L'hardware Pyboard e il software Micro Python hanno una relazione molto più interessante che le solite piattaforme. Diversamente da Arduino, non scrivete un piccolo segmento di memoria con un codice simile a C scritto su computer: potete anche comunicare direttamente con la vostra Pyboard via Micro Python usando una connessione seriale. Micro Python funge sia da pseudo sistema operativo che da linguaggio di scripting e questo rende facile e veloce provare idee o fare il debug. Pyboard si connette al vostro computer via USB come Arduino, così da poter caricare nuovi script: inserite la microUSB in una porta libera del computer (basterà una qualsiasi cavo di carica per Android). Diversamente da Arduino, tuttavia, ha un quantitativo di memoria piuttosto generoso e, collegandolo al PC, si mostra come un normale dispositivo di archiviazione. In questo modo potete scrivere uno script Micro Python usando il vostro editor di testo preferito o qualsiasi IDE e semplicemente trascinare il file prodotto sulla flash onboard o nella scheda microSD installata. Già, è veramente così semplice. Troverete quattro file per default nella Pyboard: un readme, un driver Windows (per configurare il dispositivo USB seriale), boot.py e main.py. Come in qualsiasi applicazione Python, main.py è lo script principale che contiene il vostro programma, ma non viene eseguito finché boot.py non è stato lanciato. Quest'ultimo è pensato per configurare come si comporterà la Pyboard una volta accesa. Più tardi vedrete come sfruttare questo aspetto della Pyboard per farla comportare come una comune HID (*Human Interface Device*) USB.

Hello, World!

Cominciate a prender confidenza con Micro Python (licenza MIT) e Pyboard con un semplice 'Hello, World!'. Pyboard ha quattro indicatori LED integrati, l'ultimo dei quali può essere regolato con PWM (*Pulse Width Modulation*), quindi giocateci per essere sicuri che tutto funzioni correttamente. Per prima cosa collegate Pyboard al vostro computer con un cavo MicroUSB. Nella maggior parte dei casi la memoria flash integrata in Pyboard verrà montata automaticamente come se fosse una classica memoria USB. Se la vostra distribuzione Linux richiede il mount manuale degli archivi potete semplicemente lanciare il comando **lsblk** da terminale per vedere tutti i drive connessi e quindi fare il mount con **mount /dev/sdbX** (dove sdbX è il nome del dispositivo trovato). Una volta montato, fate doppio click sul dispositivo d'archiviazione apparso e aprite **main.py** in un editor di testo. Per cominciare ci sarà un documento vuoto con una riga commentata che recita 'put your code here', quindi fatelo!

```
led = pyb.LED(4)
brightness = 0
while True:
        if brightness < 255:
```



› Non c'è molto che Pyboard non possa fare, e ha alcune gustose periferiche integrate. Potete anche controllare fino a quattro motori servo direttamente dalla board

```
                brightness += 5
                led.intensity(brightness)
                pyb.delay(50)
        else:
                brightness = 0
```

Come probabilmente capirete, questo semplice script è pensato per far passare uno dei LED di Pyboard dalla luminosità minima alla massima e viceversa, all'infinito. Sulla prima riga costruite un oggetto **led** in grado di comandare il LED PWM della board. Nella seconda riga create una variabile "luminosità" e nella terza lanciate un ciclo infinito così da modificare in eterno la luminosità. Il ciclo fa sì che, se la luminosità non è al massimo, la potete incrementare (in questo caso di 5). Quindi impostate di conseguenza l'intensità del LED e ritardate il ciclo successivo di 50ms così da vedere il cambiamento in modo graduale. Perché lo script funzioni dovete salvare il documento modificato, espellere (smontare) Pyboard e quindi premere il pulsante **RET** (*reset*) sulla board stessa. Troverete sotto alla porta microUSB, vicino al pulsante **USR** (*user*), un pulsante programmabile che incorporerete nello script fra un attimo. Se tutto va bene il LED blu tra la porta microUSB e la scheda microSD dovrebbe cominciare l'azione. Se non lo fa, avete un errore nel codice oppure Pyboard non funziona: in quest'ultimo caso provate a dialogare direttamente con la board per vedere cosa succede...

Python OS

Finora avete usato Micro Python e Pyboard allo stesso modo in cui usate una qualsiasi altro microcontroller: collegato al computer, avete scritto del codice, lo avete caricato nel controller e, si spera, lo avete visto in esecuzione. Seppure più conveniente degli altri microcontroller (non avete bisogno di un ambiente di sviluppo dedicato o driver proprietari o software perché funzioni) per tutti gli altri aspetti funziona come un comune hardware. Scopritene il vero potere collegandovi direttamente nella Pyboard per via seriale per avere un comodo prompt dei comandi. È effettivamente un Python REPL, ma più di questo è un vero sistema operativo per Pyboard: non c'è alcun aspetto che non possiate controllare da REPL (anche il file system!) Potete usare molti programmi per la connessione seriale per entrare nella REPL di Pyboard, uno su tutti **Screen**. Come potete leggere nel readme nella Pyboard, potete accedere al prompt seriale dal vostro Linux scrivendo in un terminale:

```
screen /dev/ttyAMC0
```

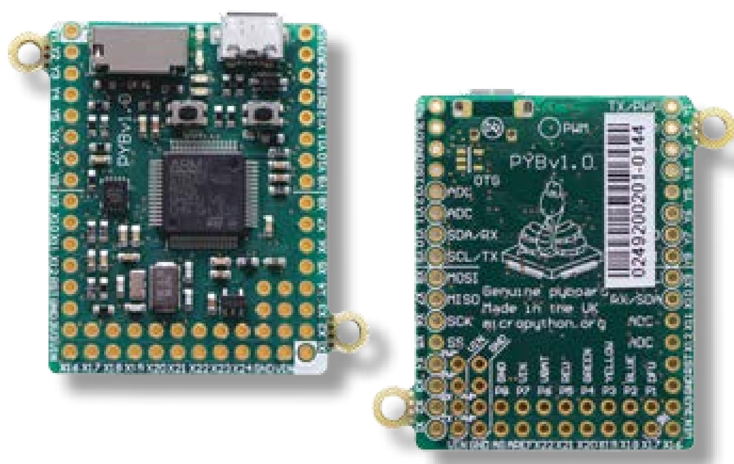
Per utenti Mac il comando è simile (**screen /dev/tty.usbmodem***) ma, come vi aspetterete, è completamente diverso per gli utenti Windows. La soluzione richiede che andiate in **Gestione dispositivi**, clicchiate con il tasto destro **Periferica sconosciuta** e aggiorniate il driver usando **pybcdc.inf** reperibile nella Pyboard stessa. Servirà anche un programma come **PuTTY** per raggiungere la REPL. Se dovete usare per qualsiasi ragione Windows, trovate le istruzioni complete all'indirizzo **http://bit.ly/windowsrepl**.

Python, parlami!

Se vi viene mostrato uno schermo vuoto, premete **Invio** per lanciare il prompt di Python, ma se ancora non succede nulla assicuratevi di non avere uno script in esecuzione sulla Pyboard. Non viene spiegato dalla documentazione, ma se uno script sta girando non potrete accedere al prompt. Il modo più veloce per risolvere è cancellare il codice da **main.py** e riavviare manualmente Pyboard usando il pulsante **RET**. Una volta raggiunta la REPL potete semplicemente 'parlare »

Tip

La cosa migliore riguardo la compatibilità delle schede microSD su Pyboard non è solo che potete scambiarle velocemente e facilmente per lanciare script Micro Python, ma potete usarle anche per registrare dati.



› Anche se più piccola di molte board Arduino (solo 33 x 40 mm), Pyboard ha un potente processore Cortex a 32 bit

in Python' con la vostra Pyboard. Se avete trovato difficoltà con lo script precedente, scoprirete che il testing è dieci volte più veloce su REPL. Provate le tre righe seguenti, premendo **Invio** dopo ogni riga:

```
led = pyb.LED(1)
led.on()
led.off()
```

Il LED rosso dovrebbe essersi acceso al secondo comando e spento dopo l'ultimo. Potete usare REPL per accedere a tutte le cose integrate in Pyboard. Potete sfruttare la libreria pub per leggere l'accelerometro, misurare la temperatura o altre periferiche collegate come i motori servo. Potete anche lanciare un reset software della Pyboard con **Ctrl**+**D** al prompt. Usando la ben conosciuta libreria **OS** di Python potete spingere l'interazione ben oltre. All'indirizzo **http://bit.ly/REPLos** troverete un corposo elenco di servizi per il sistema operativo, ma è piuttosto semplice e facile da usare.

Deliziosi quadricotteri

Secondo il suo creatore, Micro Python e Pyboard sono stati creati per fare un piccolo robot. Con l'aiuto della community, Damien George sta portando avanti un progetto per costruire un quadricottero comandato da Micro Python. Seppure ancora in fase embrionale ci sono già diverse pagine di commenti e dettagli di design che potete leggere su **http://forum.micropython.org**.



› **Damien George ha creato Pyboard grazie al suo amore per la robotica**

Per esempio, come fareste nel comune Python, potete chiamare **os.listdir()** per vedere quali file sono disponibili nella directory attuale. Assumendo che abbiate inserito una SD contenente dei file potete elencare la directory della flash integrata con **os.listdir('/flash')** o viceversa con **/sd**. Potete anche aggiungere e cancellare directory e file. Probabilmente la libreria più importante è quella disegnata per controllare la Pyboard stessa, **pyb**. Il creatore di Micro Python ha fatto l'impossibile per garantire che poteste trarre il meglio da tutto l'hardware integrato nella board e avete appena iniziato a grattarne la superficie. Per esempio, potete commutare lo stato dei LED semplicemente chiamando **.toggle()** oppure usare il pulsante **USR** come switch. Nel secondo caso, mentre il pulsante è premuto il vostro script ritornerà True, altrimenti False. Ecco un piccolo esempio per lanciarvi:

```
light = pub.LED(4)
button = pyb.Switch()

while True:
        if button():
                light.on()
        else:
                light.off()
```

Non dimenticate che se copiate questo codice nella Pyboard dal vostro PC, dovrete smontare e riavviare la board per eseguirlo. Se state usando REPL, tuttavia, potete fermare lo script con **Ctrl**+**C**, come fareste per fermare qualsiasi vecchio codice Python da riga di comando. Potete trovare l'intero ventaglio delle funzionalità di pub su **http://bit.ly/pyb_lib**, è sufficiente dire che si estende ben più al di là degli scopi di questo articolo.

Accedere all'accelerometro

Il vostro obiettivo finale per questo tutorial è trasformare la Pyboard in un simil-mouse, riuscendo a muovere il cursore inclinando la board e utilizzando le letture dell'accelerometro. Come vi aspetterete, la comunicazione con l'accelerometro integrato sulla Pyboard viene eseguita con una classe della libreria pyb. Creare e leggere da un oggetto accelerometro in REPL è semplice:

```
wobble = pyb.Accel()
while True:
        print("X:", wobble.x(), "Y:", wobble.y(), "Z:",
wobble.z())
        pyb.delay(1000)
```

Tutto quello che si fa qui è creare un oggetto accelerometro e stamparne i valori X, Y e Z per visualizzarli finché muovete la board. Continuerà a stampare risultati fintantoché scemerà l'entusiasmo e ne fermerete l'esecuzione con **Ctrl**+**C**. È stato inserito un ritardo di un secondo così che i risultati non spariscano dallo schermo prima di poterli registrare, ma potete anche ritornare una tupla filtrata con **wobble.filtered_xyz()** nell'esempio di cui sopra. Sono cose piuttosto standard, in realtà, ma qui sta la bellezza di Micro Python: gli utenti scrivono praticamente in Python 3.3. Se volete potete registrare le letture dall'accelerometro sulla scheda SD inserita per rivederle in un secondo momento, oppure potete usarle per controllare altre funzionalità della board per esempio potete accendere LED diversi a seconda dell'inclinazione della board oppure, usando la classe Servo, potete creare un robot autoguidante). Ora prenderete le letture dell'accelerometro e le inserirete in una piccola classe pensata per muovere il cursore

del mouse, chiamata **hid**. Questa funzione prende una tupla o un elenco contenente pulsanti e coordinate che rappresentano eventi HID di movimento del mouse e li gestisce. Come vedrete fra un attimo, il codice è piuttosto semplice, ma dovete saltare qualche ostacolino per far sì che il vostro computer creda che la Pyboard sia un mouse e non una periferica di archiviazione.

Tesori nascosti

Come detto poc'anzi, nel livello superiore del filesystem di Pyboard c'è **boot.py**. Questo script è la prima cosa che viene eseguita quando accendete la Pyboard ed è pensato per modificarne il comportamento. Date un'occhiata al contenuto dello script e vedrete diverse opzioni commentate. A voi interessa l'ultima, che dice che la Pyboard 'funziona come dispositivo seriale e mouse'. Se decommentate l'ultima riga, salvate il file e riavviate, non potrete più accedere al filesystem per scrivere il codice che vi serve per controllarlo. Quindi decommentate la riga e salvate boot.py ma non riavviatelo fintantoché non abbiate inserito il codice sottostante in **main.py** e lo abbiate salvato:

```
import pyb
move = pyb.Accel()
button = pyb.Switch()

while True:
        if button():
                pyb.hid((0, move.x(), -move.y(), 0))
                pyb.delay(25)
```

Qui, come già visto, avete creato oggetti per pulsante e accelerometro e realizzato un ciclo infinito con **while True**. All'interno del ciclo controllate se il pulsante è stato premuto, spostate il mouse (usando il metodo hid) verso le coordinate create dalle letture di X e Y dell'accelerometro. Il ritardo è per stabilità: sentitevi liberi di modificarlo come meglio si adatta a voi. Una volta salvato lo script potete smontare la Pyboard e riavviarla. Ora, quando premete il pulsante **USR** sulla Pyboard e lo spostate, il cursore del mouse comincerà a muoversi per lo schermo. Divertente, ma non è un vero mouse finché non potete cliccare su qualcosa. La soluzione è semplice, ma prima dovete resettare la Pyboard...

Safe Mode

Dal momento che avete cambiato la modalità in cui Pyboard si comporta quando è collegato a un computer, non potete semplicemente staccarlo e attaccarlo per accedere al filesystem così da modificare il codice. Dovete forzare la board a un riavvio in modalità sicura. Per farlo, tenete premuto il pulsante **USR** e quindi premete **RST**. Aspettate un secondo, quindi rilasciate **USR** e il LED arancio lampeggerà qualche volta in rapida successione indicando l'avvenuto reset. Ora dovreste poter accedere nuovamente al filesystem. Per far ritornare la Pyboard in modalità normale potete modificare il commento **#** davanti all'ultima riga di boot.py. Se volete continuare a usare la modalità mouse ma modificando lo script, ignorate boot.py e aprite main.py. Per far sì che la pressione del pulsante sulla Pyboard venga registrata come click del mouse, potete usare questo codice:

```
import pyb
move = pyb.Accel()
button = pyb.Switch()

while True:
        pyb.hid((button(), move.x(), -move.y(), 0))
        pyb.delay(25)
```

Stavolta quando smontate la Pyboard e riavviate, inclinare la board farà muovere il mouse automaticamente e la pressione del pulsante USR simulerà un click singolo sullo schermo. LXP

Specifiche Pyboard	
CPU Cortex M4 a 168 MHz con unità hardware per calcolo in virgola mobile	3 convertitori da analogico a digitale a 12 bit (su 16 pin)
1024 KB flash ROM e 192 KB RAM	2 convertitori da digitale a analogico (DAC) a 12 bit (su 2 pin)
Slot per scheda SD	4 LED (rosso, verde, giallo e blu)
Accelerometro a 3 assi (MMA7660)	Regolatore di voltaggio integrato da 3.3V LDO e range di voltaggio in input da 3.6V a 10V
Clock realtime con backup opzionale su batteria	Bootloader DFU in ROM per aggiornamento semplice del firmware
30 pin GPIO	

Python "micronizzato"

Anche se Micro Python è interamente nuovo ed è una implementazione separata del linguaggio Python, diverse persone puntualizzano che, dal punto di vista dell'utente, è ancora essenzialmente Python (Python 3.3 per la precisione). Il codice che inserite è lo stesso e l'output è quello che vi aspettate. Naturalmente Python è popolare perché modulare ed è stato esteso negli anni con librerie specializzate in un vasto numero di compiti. Anche se la maggior parte delle librerie sono già compatibili con Micro Python e altre richiedono solo qualche modifica, Micro Python non sarebbe particolarmente micro se contenesse tutto l'arsenale offerto dalla Standard Library.

Contiene comunque otto librerie particolarmente utili: **math**, **cmath** (per numeri complessi), **gc** per il vitale **grbage collection**, **os**, **select**, **struct**, **sys** e **time**. Gli sviluppatori hanno 'micronizzato' anche alcune parti della standard library per renderle più utili alle applicazioni Micro Python: **ujson**, **ujeapq** e **usocket**. La 'u' iniziale distingue le versioni micronizzate. Se state cercando altre librerie potete trovarle nell'archivio **micropython-lib** su **http://github.com/micropython**. Un buon numero di queste sono segnaposto e altre sono work in progress, ma se volete dar loro una possibilità o aiutare con lo sviluppo sarete sicuramente i benvenuti.



› **Andate su Github per avere info e un archivio di librerie Python compilate per Micro Python**

Creare temi personalizzati

Lo staff di Linux Pro vi spiegherà come creare il vostro tema personalizzato per **Ghost**, la famosa piattaforma di blogging Open Source


INTERMEDIO



Abbiamo parlato dei fondamenti di Ghost in un articolo precedente, ma nel caso ve lo foste perso vi ricordiamo che Ghost è una piattaforma Open Source per il blogging. Fondata con successo tramite Kickstarter nel maggio del 2013, ha superato la sua richiesta di quasi 30.000 € raccogliendone più di 196.000. Creato da John O'Nolan, Ghost ha come scopo principale quello di offrire a chi vuole creare un blog un'interfaccia semplice che permetta di scrivere e pubblicare i propri contenuti senza i problemi o le distrazioni derivanti dalla complessità dello sviluppo tramite piattaforme tradizionali. Ghost è stato progettato da zero. La sua interfaccia semplice e pulita permette di ricercare velocemente nell'archivio facendo sì che si spenda meno tempo a gestire il proprio blog lasciando più tempo per bloggare. Offre una schermata intelligente per la scrittura utilizzando l'editor Markdown con un'anteprima in tempo reale nella parte destra dello schermo e la possibilità di trascinare all'interno le immagini. Ghost segue tre principi fondamentali: primo, è sviluppato per gli utenti e non per gli sviluppatori a differenza di molte piattaforme di blogging o CMS; secondo, la piattaforma ha una licenza MIT quindi potete farne quello che volete con pochissime limitazioni; terzo, è sviluppato soltanto per amore. Ghost è un'organizzazione non-profit, il che significa che l'unica motivazione è quella di supportare i blogger e non soddisfare gli investitori. In questo tutorial vi mostreremo come installare e configurare Ghost nella vostra macchina locale e come costruire il vostro primo tema per Ghost. Per iniziare a costruire il nostro tema per Ghost, dobbiamo partire dalla directory di installazione (leggete il box **Installare Ghost**). Nella cartella **content/themes**, creiamo una nuova directory chiamata **miotema** o con il nome che preferite, assicuratevi che sia in minuscolo e senza spazi (i trattini sono ammessi). Questa directory andrà a contenere il codice del nostro tema. All'interno di questa directory creiamo i seguenti file e cartelle:

```
assets/
        css/
                normalize.css
                screen.css
        images/
        js/
        fonts/
partials/
        header.hbs
default.hbs
index.hbs
post.hbs
```

index.hbs e **post.hbs** sono gli unici file richiesti per un tema valido. Senza questi file otterrete un errore. Ora nel dashboard di Ghost andate su **Settings** -> **General**. In **Theme**, scegliete il tema chiamato **miotema**; se non lo trovate riavviate ghost. Cliccate su **Save** per attivare il tema. Non noterete modifiche di sorta per il momento. Questo ovviamente perché non abbiamo ancora inserito niente nei file del nostro tema.

Utilizzare Handlebars

Ghost utilizza un linguaggio di template chiamato **Handlebars.js** (**http://handlebarsjs.com**) e le sue espressioni predefinite rendono più semplice la creazione e il mantenimento dei temi di Ghost. Handlebars separa i template dall'HTML per voi. Inoltre, grazie a lui, è possibile scrivere funzioni o salvare variabili. Handlebars è stato sviluppato per mostrare i contenuti dove stabilito dalle espressioni. Le espressioni di Handlebars sono

Tip

Tutti i file di cui avete bisogno per questo tutorial sono reperibili all'indirizzo **http://bit.ly/1FWPE8p**.

Installare Ghost

Ghost è un'applicazione Web molto leggera. Installarla in locale e farla funzionare è questione di pochi minuti. Ghost è un'applicazione JavaScript scritta con Node.js. Potete installare quest'ultimo dal repository della vostra distribuzione, quasi certamente conterrà una versione dell'ultima serie stabile 0.10.x. In Debian/Ubuntu/Mint ciò può essere fatto con:

```
$ sudo apt-get install nodejs
```

Se volete una versione diversa potete scaricarla direttamente dal sito **http://nodejs.org**, esiste anche una guida in inglese per l'installazione in praticamente tutte le distribuzioni disponibili (**http://bit.ly/InstallingNodejs**).
Ufficialmente Ghost non è stato incluso nei pacchetti di molte distribuzioni, quindi andate all'indirizzo **https://ghost.org/download** per scaricare l'ultima versione e decomprimetela. Per installare Ghost eseguite da terminale come root i seguenti comandi:

```
$ cd /percorso/del/download/ghost
$ npm install --production
```

Un gestore di pacchetti javascript chiamato **npm** (che normalmente viene installato con Node.js ma che comunque sarà sicuramente disponibile nel repository di sistema) provvederà a installare tutte le dipendenze necessarie. Una volta completato potrete avviare Ghost in modalità sviluppo con:

```
$ npm start
```

Ora nel vostro browser potrete accedere alla vostra installazione di Ghost andando all'indirizzo **http://127.0.0.1:2368**, e potrete accedere all'amministrazione all'indirizzo **http://127.0.0.1:2368/ghost**. Quando visitate questo indirizzo per la prima volta vi verranno richieste le informazioni per creare il vostro blog: il nome del blog, il vostro nome, la vostra email e una password.

racchiuse tra parentesi graffe e somigliano a: {{author.name}}. Sostanzialmente questo prende la proprietà **author.name** e la mostra. Vediamo di sporcarci le mani e iniziare a creare il nostro tema. Apriamo il file **default.hbs** con un editor di testo. Questo file contiene il template principale e include tutti i tag di base <html>, <head>, <body> che verranno usati nel sito Ghost. In questo template inseriamo il nostro doctype HTML, i meta tag e i tag di head e body. (Prendete il contenuto dal file **default.hbs** reperibile all'indirizzo **$indirizzocodice$**). Noterete delle espressioni come: {{! Meta Tags Responsive }}. Tutte le espressioni anticipate da un punto esclamativo dentro le parentesi graffe sono commenti e non vengono processati. L'espressione **{{ghost_head}}** è usata per mostrare tutti gli script di sistema, gli stili e i meta tag. **{{ghost_foot}}** è utilizzato per mostrare gli script in fondo al documento. L'espressione **{{{body}}}** è molto importante. Qui è dove tutto il contenuto viene mostrato, estendendo il template di default. Il tag **{{body_class}}** viene utilizzato per generare il nome della classe CSS in base alla pagina:

```
<body class="{{body_class}}"> <div class="miotema_page">
{{{body}}} </div>
{{ghost_foot}} </body>
Index.hbs
```

Ora diamo un'occhiata al nostro **index.hbs** (sempre preso dall'indirizzo di prima), qui vediamo l'espressione {{!< default}} in testa al documento per richiamare il template di base visto prima. Questo template verrà utilizzato dalla nostra homepage. Possiamo impostare lo stile di ogni post del blog all'interno dell'espressione foreach. All'interno del ciclo che inizia con {{#foreach posts}} e finisce con {{/foreach}} vediamo il codice che verrà utilizzato per ogni post. Per mostrare il contenuto di ogni post del blog utilizziamo {{content}}. Possiamo limitare il numero di parole visualizzate utilizzando words="100". Come potete notare tutti i nomi delle classi iniziano con **miotema_**. È una pratica consigliata quando si costruisce un tema Ghost. Ghost assegna in automatico nomi e ID particolari a certi elementi del nostro tema; è sempre meglio evitare di rischiare di utilizzare nomi che potrebbero venire utilizzati dal sistema.



› **Potete usare l'espressione {{date}} per visualizzare la data di pubblicazione e l'opzione format per controllarne il formato**

Partials

Normalmente possiamo inserire il nostro markup di header subito sotto a {{!< default}}, ma il vantaggio dei template Handelbars è che supportano le gerarchie e un template può estenderne un altro. Questo include l'utilizzo di parti. Questa prassi aiuta a evitare ripetizioni e incoraggia il riutilizzo del codice. Possiamo dividere il nostro header in un template parziale. Prendiamo dal solito indirizzo anche il contenuto del nostro file **header.hbs** nella directory **partials**.
Nel menu **Settings** di Ghost, possiamo caricare il logo del nostro blog e la sua immagine di copertina. Come potete vedere nel template testiamo la presenza dell'immagine di copertina. Se esiste, la mostriamo come immagine di sfondo:

```
{{#if @blog.cover}}
        style="background-image: url({{@blog.cover}})"
{{/if}}
```

Allo stesso modo controlliamo che esista il logo.

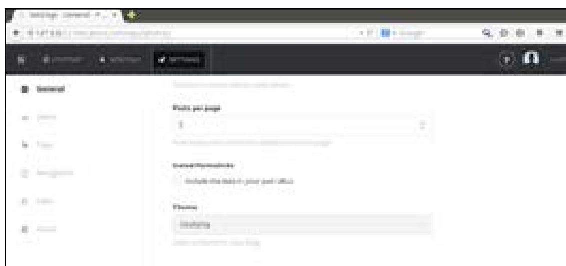
```
{{#if @blog.logo}}
        <a class="blog-logo" href="{{@blog.url}}">
                <img src="{{@blog.logo}}" alt="Logo del
blog" />
        </a>
    {{/if}}
```

La variabile globale **@blog** ci da accesso alle impostazioni globali di Ghost che possiamo mostrare nel nostro tema.
Ora tuffiamoci nella personalizzazione del nostro tema. All'interno del nostro template abbiamo incluso **normalize.css** che effettua un po' di pulizia sui CSS. Dentro a **screen.css** invece, andiamo a mettere tutti i nostri stili personalizzati. Aggiungiamo alcuni stili generali, per esempio impostiamo il **max-width** per far sì che il nostro layout non superi le dimensioni che vogliamo: »

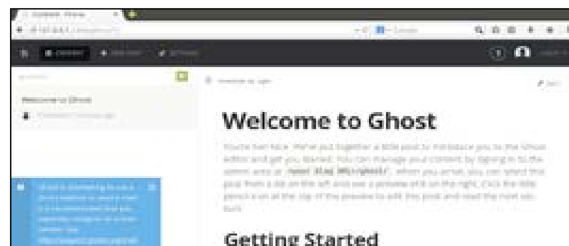
```
.miotema_page {
		max-width: 980px;
		margin: 0 auto;
}
.miotema_header {
		padding:20px 0;
		text-shadow: 2px 2px 2px rgba(26, 26, 26, 0.1);
		text-align: center;
		color: #2F3727;
}
```

Ora diamo uno stile a ogni post del blog e al suo contenitore chiamato article:

```
main article {
		margin: 30px 0;
		border-left:1px solid #DBDBDB;
		border-right:1px solid #DBDBDB;
		background-color: #FFFFFF;
}
.miotema_post_content {
		padding: 0 20px;
}
.miotema_post_title {
		margin: 0 0 20px 0;
		padding: 10px;
		font-size: 2em;
		letter-spacing: 2px;
		text-align: center;
		text-shadow: 2px 2px 2px rgba(26, 26, 26, 0.2);
		color: #FFFFFF;
		background-color: #8ACD36;
}
.miotema_post_title a {
		text-decoration: none;
		color: #FFFFFF;
}
.miotema_main_img img {
		width: 100%;
		max-width: 100%;
		border: 0;
}
```



Attiviamo il nostro nuovo tema dalla pagina di amministrazione. Potrebbe essere necessario un restart di Ghost da terminale perché il tema compaia nell'elenco



Dopo l'installazione Ghost vi consiglia di configurare un servizio per le email invece di utilizzare l'invio diretto

Piazziamo la data nel lato sinistro e il pulsante per aprire il dettaglio del post dall'altra parte. Facciamo sì che il link abbia l'aspetto di un pulsante

```
.miotema_post_info {
		overflow: auto;
		padding: 0 20px;
		background-color: #98C148;
}
.miotema_date {
		float: left;
		padding-top: 20px;
		color: #FFFFFF;
}
.button {
		float: right;
		padding: 20px 0;
}
.button a {
		padding: 5px;
		transition: ease .3s;
		text-decoration: none;
		color: #FFFFFF;
		background-color: #39a9da;
}
.button a:hover {
		background-color: #199ed9;
}
```

Abbiamo toccato gli aspetti chiave dello sviluppo di un tema per Ghost. Speriamo che questo articolo possa aver stimolato la vostra fantasia e vi abbia offerto gli elementi necessari a realizzare il vostro tema personale. Scoprirete che personalizzare il proprio tema è un gioco da ragazzi grazie alle espressioni di Handlebars. Ancora una volta vi ricordiamo che tutto ciò che vi serve lo potete trovare all'indirizzo **http://bit.ly/1FWPE8p**. Per maggiori approfondimenti e per conoscere tutto ciò che c'è da sapere sui temi di Ghost trovate la documentazione ufficiale all'indirizzo **http://docs.ghost.org/themes/**. LXP

Preparare post.hbs

Abbiamo visto soltanto il template per l'homepage ma il file **post.hbs** si occupa della visualizzazione del singolo post. Anche in questo template (di nuovo potete trovare il file **post.hbs** al solito indirizzo **http://bit.ly/1FWPE8p**) abbiamo utilizzato le espressioni di Handelbars {{!< default}} e {{> header}}. Questa volta abbiamo utilizzato le espressioni {{#post}} e {{/post}} per mostrare i dati del singolo post. All'interno del singolo post abbiamo accesso ai dati dell'autore compreso il nome e la biografia. Possiamo mostrare le informazioni sull'autore aggiungendo semplicemente questo codice:

```
<section class="author">
		<h3>Scritto Da:</h3>
		<h4>{{author.name}}</h4>
		<p>{{author.bio}}</p>
	</section>
```

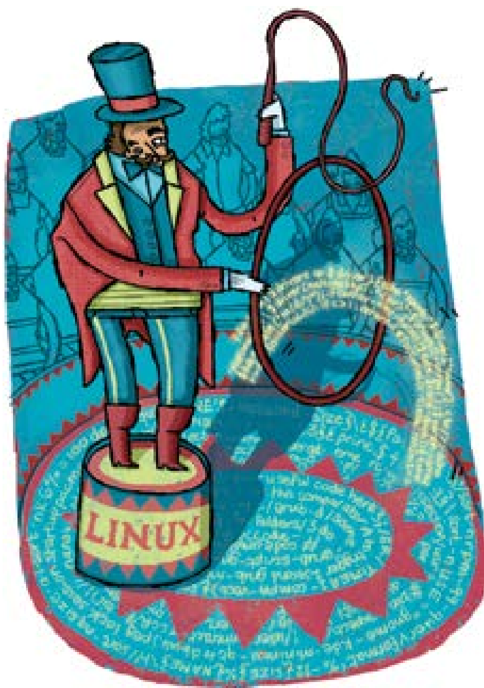
Ora possiamo aggiungere un po' di stili CSS al nostro post. È sempre meglio inserire gli stili all'interno di un unico file CSS separato. È importante perché, in un prossimo futuro, Ghost svelerà una nuova funzionalità a schermo diviso, che caricherà tutti gli stili personalizzati del tema nell'interfaccia di admin.

Convertire le immagini

Lo staff di Linux Pro vi insegnerà come dimenticarvi di Gimp, buttare via il mouse e imparare a modificare le immagini da riga di comando



INTERMEDIO



Chiedete a un gruppo di utenti Linux di farvi il nome di un pacchetto di elaborazione grafica e la stragrande maggioranza vi nominerà **Gimp** (*GNU Image Manipulation Program*). Potreste ottenere qualche voto per altri programmi ma purtroppo non saranno in molti a citare uno dei pacchetti grafici più utilizzati in Linux, **ImageMagick**. Si tratta di una suite di programmi per l'*image processing* da riga di comando. Potreste chiedervi quale sia l'utilità di una cosa del genere. Sicuramente la manipolazione delle immagini è più facile da gestire con un'applicazione grafica. Se volete ritoccare o modificare una singola immagine, sicuramente è così, ma se invece dovete cambiare formato o ridimensionare un'intera cartella di immagini? Qui è dove entra in gioco ImageMagick e questo è il motivo per cui è uno dei tool di manipolazione delle immagini più utilizzato in Linux, anche se lo fa dietro le quinte. Il comando più utilizzato di ImageMagick tra quelli disponibili è **convert**, e cambiare il formato di un'immagine è l'operazione più semplice che è in grado di fare. Lanciare il comando

```
convert pic1.jpg pic1.png
```

farà esattamente quello che ci si può aspettare, converte un immagine da JPEG a PNG. In realtà, crea un'immagine nuova, non modifica nulla. Non è necessario specificare il tipo di input e output, **convert** capisce il formato di input dal file stesso (quasi per magia) e utilizza l'estensione per scegliere il formato dell'output. Potete comunque impostare manualmente il formato inserendo prima del nome del file di output il tipo di immagine seguito da due punti, in questo modo:

```
convert pic1.jpg bmp:pic1.gif
```

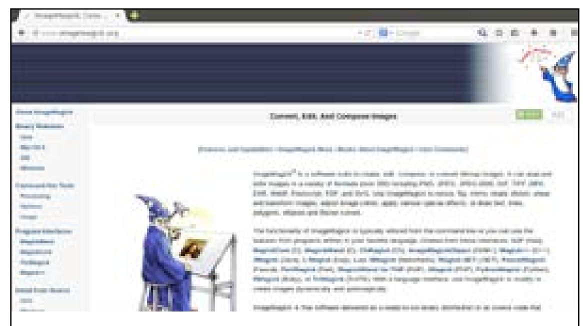
Questo creerà un file BMP con estensione GIF: può sembrare inutile, ma considerate che come molti altri programmi **convert** permette di passare i dati allo standard input/output utilizzando come nome file -. Quindi se volete inviare i dati di un'immagine a un altro programma in formato PPM potete farlo così:

```
convert pi1.jpg ppm:- | altro_programma
```

Alcuni formati di output accettano anche opzioni aggiuntive, come per esempio l'impostazione della qualità del JPEG:

```
convert pic2.png -quality 75 pic2.jpg
```

La man page di convert e la documentazione online vi offrono l'elenco completo di queste opzioni.



› **Se trovate la man page difficile da leggere, il sito Web di *ImageMagick* vi offre molti dettagli ed esempi di comandi**

Ridimensionare le immagini

Dopo il cambio di formato, il ridimensionamento è una delle principali operazioni che si svolgono sulle immagini, e **convert** fa anche questo.

```
convert pic3.png -resize 50% pic3_piccola.png
```

Questa è l'operazione di ridimensionamento più semplice, tutta l'immagine viene ridotta della metà. Potete ridimensionare anche utilizzando fattori diversi per gli assi X e Y con **40x60%**, anche se c'è un solo simbolo % si applica a entrambi i numeri. Se viene fornito un unico valore numerico viene considerato come la larghezza in pixel della nuova immagine; l'altezza viene

calcolata di conseguenza per mantenere le proporzioni dell'immagine originale.

```
convert pic3.png -resize 1000 pic1000.png
convert pic3.png -resize x1000 pic1000.png
```

Nel secondo esempio, il valore è per l'altezza, pensatela come larghezzaXaltezza solo che manca la larghezza. Se vengono passati sia la larghezza che l'altezza verrà creata un'immagine il più grande possibile all'interno di quelle dimensioni mantenendo le proporzioni originali. Se volete impostare delle dimensioni assolute ignorando le proporzioni potete aggiungere **!** alle dimensioni:

```
convert pic4.png -resize 1000x600 pic5.png
convert pic4.png -resize 1000x600! pic5.png
```

La prima riga preserva le proporzioni, la seconda crea un'immagine esattamente di 1.000x600 pixel. Se state creando delle anteprime di immagini più grandi, potreste voler evitare di allargare eventuali immagini più piccole, visto che questo creerebbe un'immagine più grande ma meno definita, per farlo vi basta aggiungere **>** in questo modo:

```
convert pic4.png -resize 1000x600\> pic5.png
```

Il **>** dice a **convert** di ridimensionare l'immagine soltanto se l'originale è più grande delle dimensioni date. Da notare che dovete fare l'escape del simbolo > quando lo eseguite nella shell per evitare che venga interpretato come operatore di ridirezionamento. Diverse operazioni possono essere combinate in un'unica chiamata, e vengono applicate in ordine, quindi potete ridimensionare e convertire a un formato diverso allo stesso tempo:

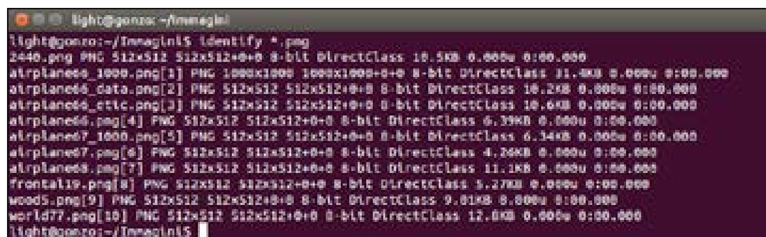
```
convert pic6.png -resize 30% -quality 75 pic6a.jpg
```

Aggiungere del testo

Un'opzione interessante è la possibilità di aggiungere del testo sopra a un'immagine, per esempio:

```
convert pic7.png -gravity SouthEast -family Times -pointsize 30
-annotate +10+10 “Del testo” etichetta7.png
```

Il componente chiave qui è annotate, seguito dalla posizione x,y del testo e dal testo stesso. Anche l'impostazione **-gravity** è importante, determina da dove devono venire calcolate le coordinate. In questo esempio il testo è disegnato a 10 pixel di distanza dall'angolo in basso a destra dell'immagine. Le opzioni **-family** e **-pointsize** controllano il font e le dimensioni del testo. È sempre meglio utilizzare **-family** invece di **-font** perché **-family** utilizzerà il font più simile nel caso in cui il font esatto non fosse disponibile, mentre **-font** da errore. Le opzioni di ImageMagick si applicano nell'ordine in cui sono scritte quindi è obbligatorio specificare il font e la direzione della gravità prima dell'opzione **-annotate** altrimenti non verranno applicati



› **Potete usare identify per ottenere informazioni aggiuntive su file di immagini di ogni tipo**

(a meno che non ci sia un'altra operazione di testo dopo, in quel caso verranno applicati a quella). Potete utilizzare il comando **date** per aggiungere la data di creazione all'immagine, così:

```
convert pic7.png -gravity SouthEast -annotate +10+10 “$(date
-r pic7.png +%d-%b-%y\ %H:%M)” nuovo7.png
```

Convert crea un nuovo file ma a volte potreste voler modificare un file esistente, ImageMagick offre il comando **mogrify** proprio per questo. Funziona allo stesso modo di **convert**, a parte che **mogrify** sovrascrive il file di input applicando i cambiamenti, confrontiamo questi due comandi:

```
convert pic8.png -resize 50% pic9_piccola.png
mogrify -resize 50% pic9.png
```

Dato che questo comando distrugge il file originale dovete assicurarvi che l'operazione che state per eseguire sia veramente ciò che volete, per questo conviene sempre testarla prima con **convert**. Comunque, **mogrify** ha un'altra freccia al suo arco. Con **convert** dovete specificare sia il file di input sia quello di output, quindi ogni cosa può essere eseguita soltanto su un singolo file. **mogrify**, invece, necessita soltanto di un nome, quindi potete processare diversi file con un solo comando, in questo modo:

```
mogrify -resize 50% *.png
```

questo comando processerà tutti i file con estensione png all'interno della cartella. Abbiamo detto che **mogrify** sovrascrive gli originali, ma c'è un'importante eccezione; se il comando viene utilizzato per cambiare il formato del file viene creato un nuovo file con l'estensione desiderata e il file originale non viene toccato, anche se vengono eseguite altre operazioni

```
mogrify -resize 25% -quality 70 -format jpg *.png
```

Questo comando creerà delle miniature JPEG di tutti i file PNG presenti nella cartella. Abbiamo visto soltanto la punta dell'iceberg di ciò che può fare ImageMagick. Per esempio, **Identify** fa ciò che promette, vi dice il formato e le dimensioni di un file, il comando **import** può essere utilizzato per prendere immagini dal server X. La man page è abbastanza difficile da leggere ma esistono spiegazioni ed esempi sul sito Web ufficiale di ImageMagick all'indirizzo **www.imagemagick.org**. LXP

Batch processing

Anche se mogrify può lavorare con diversi file, convert funziona su un singolo file alla volta, quindi come possiamo fare se vogliamo processare una directory piena di file? Il modo più semplice è di utilizzare l'operatore shell **for**.

```
for i in *.png
do
    convert “$i” -resize 500x500 “miniature/${i%.
png}.jpg”
done
```

Questo comando esegue il codice compreso tra **do** e **done** una volta per ogni file che corrisponde alla ricerca. Ogni volta, **$i** viene sostituito dal nome del file. La parte **${i%.png}** viene rimpiazzata dal nome, togliendo .png dalla fine (**%** rimuove la stringa seguente) e aggiunge l'estensione .jpg, quindi ciò che viene eseguito è simile a:

```
convert “pic1.png” -resize 500x500 “miniature/
pic1.jpg”
convert “pic2.png” -resize 500x500 “miniature/
pic2.jpg”
...
```

Racchiudiamo i nomi dei file tra virgolette nel caso in cui ce ne sia qualcuno che contiene degli spazi, che altrimenti confonderebbe la shell. In alternativa per raggiungere lo stesso scopo possiamo utilizzare **find** in questo modo:

```
find foto -name “*.jpg” -exec convert “{}” -resize
25% “miniature/{}” \;
```

Ogni file trovato viene passato al comando **-exec**, dove **{}** viene sostituito dal nome del file. Ancora una volta, utilizziamo le virgolette per evitare problemi con gli spazi.

Android: tutti i segreti della shell

Chi avrebbe mai detto che si possa imparare Linux con un semplice "telefono"? Ecco come farlo grazie ai nostri consigli...



La shell è lo strumento principe di sistemisti GNU/Linux e *power user*, e da sempre rappresenta croce e delizia per ciascun utente del pinguino: potente acceleratore di operazioni nelle mani di un utente esperto, può trasformarsi in un vero e proprio incubo se affrontata senza la giusta preparazione. A supporto della triste fama che si è costruita intorno alla riga di comando, spesso utilizzata come scusa dai più pigri (e disinformati...) come scusa per rimandare il passaggio a GNU/Linux, occorre notare che, effettivamente, l'apprendimento dei segreti della shell richiede tempo e dedizione: due fattori che non sempre vanno d'accordo con i ritmi della vita quotidiana, specie per chi utilizza il Pinguino fuori dall'ambito prettamente professionale. Ma se, da un lato, con l'esplosione delle distribuzioni e il susseguirsi di versioni sempre più *user friendly*, il peso della shell nell'amministrazione domestica si è via via alleggerito, dall'altro la crescente diffusione di uno strumento ormai comune come lo smartphone ha aperto scenari formativi sinora sconosciuti. Per quanto possa sembrare strano, infatti, uno smartphone equipaggiato con una versione recente di Android può trasformarsi in un potente alleato di ogni utente del pinguino, consentendo di studiare ed approfondire i principali comandi di shell senza dover accedere a un computer. Le app Android che descriveremo in queste puntate, infatti, ci consentiranno di emulare un terminale sullo smartphone, dotando il "telefono" di un buon set di comandi, sufficientemente rappresentativo delle reali esigenze di un utente domestico. Di fronte alla possibilità di sfruttare un qualsiasi momento di pausa per imparare o affinare la conoscenza dei principali comandi di shell, direttamente sul nostro inseparabile smartphone, quale motivo (a esclusione della nostra atavica pigrizia, naturalmente!) può impedirci di acquisire la necessaria padronanza con il terminale?

Installazione di terminale e tastiera

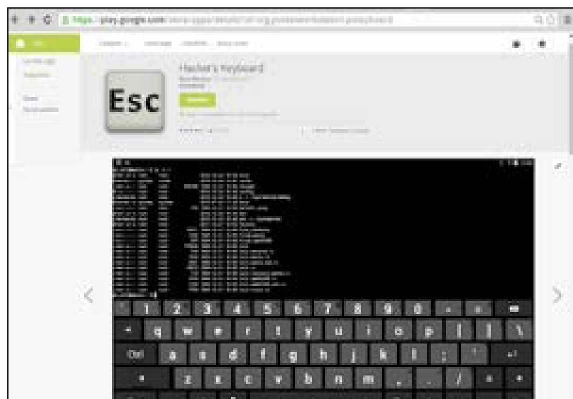
Il primo passo, ovviamente, non può che essere l'installazione di un terminale. L'app **Terminal Emulator for Android**, disponibile gratuitamente su Google Play (**Fig 1**) consente (come promesso dal nome) di accedere a un terminale GNU/Linux sul nostro smartphone. In poco meno di 1 MB, chiunque può disporre di un terminale pronto per l'uso, sebbene dotato di un set di comandi abbastanza scarno (ma che, come vedremo nel proseguo, può essere ampliato per abbracciare gran parte dei comandi di uso comune). Una volta installata l'app, tuttavia, è opportuno effettuare una seconda operazione prima di avviare la shell: scegliere una tastiera alternativa. La tastiera offerta nativamente da Android, infatti, sebbene costituisca un imprescindibile ausilio per compiere le operazioni più comuni sullo smartphone (è perfetta per telefonare, scrivere un messaggio o una email, navigare, ecc.), è pensata per uno scopo decisamente differente dall'amministrazione di una Linux box, e risente pertanto di una certa scomodità se impiegata all'interno del terminale. Chi lo desidera può migliorare sensibilmente la propria esperienza utente con l'emulatore di terminale, installando una tastiera pensata proprio con un occhio all'amministrazione dei sistemi: la **Hacker's Keyboard**, che riproduce sul nostro smartphone il layout di una normale tastiera da pc. L'app è gratuita e disponibile, al pari del terminale, su Google Play (**Fig 2**). Una volta installata, tuttavia, la tastiera deve essere abilitata dai menu di configurazione di Android: per motivi di sicurezza, infatti, il sistema richiede che ogni modifica delle modalità di input dello smartphone avvenga manualmente a cura dell'utente, negando la possibilità di modifiche automatiche da parte delle app. Se tutto ciò può rappresentare un onere aggiuntivo per un aspirante mago della shell, dall'altro costituisce una misura di sicurezza necessaria contro app malevole: cosa succederebbe se un attaccante riuscisse a installare un'applicazione (magari con un look & feel identico a quella della tastiera nativa) che, imponendosi automaticamente come tastiera di



› **Fig 1: La pagina dell'emulatore di terminale su Google Play, da cui è possibile installare l'app gratuitamente**

L'autore

Maurizio Russo
Laureato in Informatica presso l'Università "La Sapienza" di Roma, con una tesi sperimentale sullo stack TCP/IP del kernel Linux, è un utente del pinguino dal 2001. Nella sua carriera si è occupato di formazione, sicurezza, networking, progettazione e sviluppo di software.



› **Fig 2: La pagina dell'app *Hacker's Keyboard* su Google Play, da cui è possibile effettuare l'installazione (senza oneri per l'utente!)**

default dello smartphone, sia in grado di registrare ogni dato digitato dall'utente e inviarlo a un sito esterno, ivi compresi i numeri delle nostre carte di credito? Naturalmente, delegare la scelta all'utente non significa garantirne l'immunità da truffe come quelle appena descritte, ma di certo l'esame critico da parte di un essere umano può costituire un ostacolo insormontabile per app fraudolente: non a caso, a ogni modifica dei dispositivi di input da parte dell'utente dello smartphone (attraverso i menu **Impostazioni** -> **Sistema** -> **Lingua e immissione** -> **Tastiera e metodi di immissione**), Android visualizza un messaggio di warning, segnalando il potenziale pericolo che l'azione può rappresentare.
Nel nostro caso, una volta superato l'avviso e provveduto ad aggiungere la Hacker's Keyboard tra i dispositivi di input abilitati, è necessario selezionare la tastiera come dispositivo di input corrente, sancendo così la sostituzione della tastiera di Google: è sufficiente avviare l'app e premere il pulsante **Set input method**, avendo cura di selezionare la **Hacker's Keyboard** nel menu successivamente visualizzato da Android. Al termine delle nostre sessioni addestrative sulla shell, optando per la tastiera di Google nel medesimo menu potremo ripristinare lo stato originario del sistema.

Primi passi con il terminale

Terminate le predisposizioni iniziali, possiamo dedicarci alla shell vera e propria. Avviamo l'emulatore di terminale, avendo cura di tenere lo smartphone in posizione longitudinale (e lo scorrimento automatico attivato). In questa posizione è possibile, specie sugli schermi più grandi, godere appieno dei vantaggi offerti dalla tastiera appena installata, come la disponibilità dei tasti **CTRL**, **ESC**, **ALT** e **Fn** e, soprattutto del tasto **TAB** e delle frecce direzionali, senza i quali l'interazione con la shell può divenire un'esperienza a dir poco frustrante: immaginate per un attimo di non avere la funzionalità di completamento automatico offerta dal TAB, oppure di essere costretti a digitare nuovamente l'intera stringa di un comando per via di un carattere sbagliato o di uno spazio omesso... Per fortuna, l'installazione della tastiera ci ha messo al sicuro da questi inconvenienti, quindi possiamo passare subito al primo comando da esaminare. In che directory ci ha posizionato il terminale all'avvio? Scopriamolo con il celeberrimo **ls**, per visualizzare tutti i i contenuti della directory corrente:

```
ls -l
```

ci restituisce, come output, la lista di tali contenuti (**Fig 3**). Per

```
u0_a92@a50ne:/ $ ls -l
drwxr-xr-x root     root              2015-03-29 07:23 acct
drwxrwx--- system   cache             2015-03-29 15:09 cache
dr-x------ root     root              2015-03-29 07:23 config
lrwxrwxrwx root     root              2015-03-29 07:23 d -> /sys/kernel/debug
drwxrwx--x system   system            2015-03-29 07:23 data
-rw-r--r-- root     root          194 1970-01-01 01:00 default.prop
drwxr-xr-x root     root              2015-03-29 07:23 dev
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@android -> /dev/bloc
k/mmcblk0p5
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@cache -> /dev/block/
mmcblk0p6
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@ebr1 -> /dev/block/m
mcblk0p1
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@fat -> /dev/block/mm
cblk0p8
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@protect_f -> /dev/bl
ock/mmcblk0p2
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@protect_s -> /dev/bl
ock/mmcblk0p3
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@sec_ro -> /dev/block
/mmcblk0p4
lrwxrwxrwx root     root              1970-01-01 01:00 emmc@usrdata -> /dev/bloc
k/mmcblk0p7
-rw-r--r-- root     root          127 1970-01-01 01:00 enableswap.sh
lrwxrwxrwx root     root              2015-03-29 07:23 etc -> /system/etc
-rw-r--r-- root     root            0 1970-01-01 01:00 factory_init.project.rc
```

› **Fig 3: Il comando *ls* consente di visualizzare tutti i contenuti della directory corrente**

ciascun file/subdirectory contenuta nella directory corrente, ls elenca le seguenti informazioni, distinte su colonne differenti:

- tipo e permessi di accesso (vedi box **Permessi di accesso ai file**);
- utente proprietario;
- gruppo proprietario;
- dimensione in byte;
- tempo di ultima modifica (box **I tempi di accesso dei file**);
- nome.

Esaminando l'output di ls, possiamo dedurre che ci troviamo nella root (/) del dispositivo: l'invocazione del comando **pwd**, che restituisce il nome della directory corrente, ce ne fornisce la conferma definitiva (**Fig 4**). A meno di modifiche apportate allo smartphone, la root directory è, al pari di quanto avviene in una Linux box per gli utenti non privilegiati, un'area del filesystem accessibile in sola lettura. Per proseguire i nostri esperimenti, è dunque necessario spostarci in una directory sulla quale l'utente corrente abbia i diritti di scrittura, come per esempio la directory nella quale è montata la scheda sd del telefono. Come scoprirne il nome? Possiamo ricorrere a un altro comando offerto dal nostro terminale, **df**, che consente di ottenere un elenco di tutti i filesystem attualmente montati sul sistema. Il comando:

```
df
```

produce l'output rappresentato in (**Fig 5**), dal quale è possibile evincere, per ciascun filesystem elencato, i seguenti elementi:

- directory di mount (colonna Filesystem);
- dimensione complessiva in MB (colonna Size);
- quantità di spazio su disco effettivamente utilizzata (colonna Used) e quella libera (colonna Free); »

```
u0_a92@a50ne:/ $ pwd
/
u0_a92@a50ne:/ $
```

› **Fig 4: Il comando *pwd* restituisce il nome della directory corrente**

Fig 5: Il comando df all'opera: ecco i filesystem montati sullo smartphone

```
u0_a92@a50ne:/ $ df
Filesystem              Size    Used    Free    Blksize
/dev                    232M     52K    231M    4096
/mnt/secure             232M      0K    232M    4096
/mnt/asec               232M      0K    232M    4096
/mnt/obb                232M      0K    232M    4096
/system                 689M    654M     35M    4096
/data                     1G      1G    445M    4096
/cache                  124M      4M    119M    4096
/protect_f                8M      4M      4M    4096
/protect_s                8M      4M      4M    4096
/mnt/cd-rom              17M     17M      0K    2048
/storage/sdcard0         14G      4G     10G    32768
/mnt/secure/asec: Permission denied
/storage/sdcard1          1G    170M    970M    16384
u0_a92@a50ne:/ $
```

» la dimensione del blocco, ovvero l'unità minima di allocazione su disco (colonna Blksize).
Osservando attentamente l'output del comando, possiamo notare due directory che sembrano far riferimento a una sd card: la directory **/storage/sdcard1 e la directory /storage/sdcard0**. Accediamo ora a quest'ultima, con il comando:

```
cd /storage/sdcard0
```

(il carattere '/' si ottiene tenendo premuto il pulsante **7** della tastiera installata), che serve per l'appunto per spostarsi da una directory all'altra. Per i più pigri, di certo spaventati dalla lunghezza del comando da digitare, è possibile ricorrere all'autocompletamento offerto dalla nostra tastiera alternativa: dopo aver scritto la stringa 'cd /st', il nome della directory 'storage' può essere ottenuto semplicemente premendo il **TAB**; una seconda pressione del medesimo tasto farà apparire invece gran parte del nome della directory annidata (sdcard0), cui dovremo aggiungere solamente lo 0 finale, prima di confermare con il tasto **INVIO**. Una volta ultimato lo spostamento nella directory di montaggio della sdcard, è opportuno creare una directory ad hoc al cui interno convogliare i nostri esperimenti, a salvaguardia dai danni ai contenuti dello storage imputabili a eventuali operazioni maldestre. Il comando:

```
mkdir lxp
```

crea una directory di nome lxp, proprio all'interno della sdcard. Per spostarci nella directory appena creata, è sufficiente il già citato comando

```
cd lxp
```

Alla creazione, la directory è ovviamente vuota; per creare un file, in assenza di un editor fruibile da shell (lacuna che, come vedremo, sarà colmata nel proseguo), possiamo ricorrere al comando **echo**, e ai metacaratteri di redirezione dell'input/output, entrambi supportati dall'emulatore di shell. Il comando echo consente di stampare a video una stringa: per esempio, digitando

```
echo buongiorno lettori di Linux Pro
```

si ottiene la stampa a video della frase passata come

```
u0_a92@a50ne:/ $ echo $SHELL
/system/bin/sh
u0_a92@a50ne:/ $
```

Fig 6: Stampando con echo la variabile d'ambiente *$SHELL*, possiamo conoscere quale shell stiamo utilizzando: nel nostro caso, la *Bourne shell*

argomento al comando. Potrebbe sembrare una funzionalità di poco conto, ma si rivela molto utile negli script (per la stampa di messaggi per l'utente) e per la visualizzazione dei contenuti delle variabili d'ambiente (vedi box omonimo).
A tal riguardo, digitando per esempio

```
echo $SHELL
```

possiamo ottenere l'indicazione della shell attualmente in uso, che nel nostro caso risulta essere la Bourne Shell, **/system/bin/sh** (**Fig 6**). Un'altra modalità d'impiego del comando consiste nella creazione e/o modifica di file di testo, mediante l'utilizzo congiunto con i citati operatori di redirezione dell'input/output: tali operatori, infatti, consentono di "spostare" su file l'output di un comando arbitrario, sottraendolo al video cui questo è normalmente destinato. Per esempio, con

```
echo buongiorno a tutti > buongiorno.txt
```

si redirige l'output di echo verso il file buongiorno,txt (che, poiché non esiste all'interno della nostra directory di lavoro, sarà contestualmente creato), così come con

```
echo buongiorno o buonasera, a seconda di che ora sia >> buongiorno.txt
```

si provvede ad accodare (operazione detta di "append") la frase "buongiorno o buonasera, a seconda di che ora sia" al predetto file. Possiamo avere conferma del buon esito delle operazioni appena effettuate ricorrendo al comando **cat**, per la stampa dei contenuti di un file di testo. Il comando

```
cat buongiorno.txt
```

produce il risultato in **Fig 7**, dimostrando come le due stringhe siano state effettivamente inserite all'interno del file.
Reiterando il procedimento, possiamo creare un secondo file, denominato salve.txt:

```
echo salve, lettori di Linux Pro! > salve.txt
```

e un terzo, con nome saluti

```
echo in conclusione saluti a tutti, qui a  linux pro! > saluti
```

per poi leggere i contenuti di entrambi con

```
cat salve.txt saluti
```

A questo punto disponiamo di tre file nella directory, come ci può confermare il comando

```
ls -l
```

Si tratta di tre file di testo, facilmente distinguibili in base al contenuto (a patto di poterlo visionare con il comando cat).

Permessi di accesso ai file

In ambiente GNU/Linux, a ciascun file è associato un utente proprietario e un gruppo proprietario: in ambiente domestico essi corrispondono, per un file creato da un utente non privilegiato, con l'utente della macchina e il gruppo omonimo. Ciò consente di partizionare i diritti di accesso al file degli utenti in tre classi:

» i diritti del proprietario;
» i diritti del gruppo proprietario;
» i diritti di tutti gli altri utenti.

A ciascuna di queste classi è possibile associare uno o più dei seguenti diritti di accesso al file:

» diritto di accesso in lettura (r);
» diritto di accesso in scrittura (w);
» diritto di accesso in esecuzione (x).

```
u0_a92@a50ne:/storage/sdcard0/lxp2 $ cat  buongiorno.txt
buongiorno a tutti
buongiorno o buonasera, a seconda di che ora sia
u0_a92@a50ne:/storage/sdcard0/lxp2 $ |
```

› Fig 7: Esempio di utilizzo del comando *cat*, per la visualizzazione del file di testo buongiorno.txt

Supponiamo, tuttavia, di avere una moltitudine di file (facciamo 100), e di non ricordare quali abbiano qualche riferimento al tema "Linux Pro". Esiste un'alternativa più rapida al processo di visualizzazione in serie di tutti i file? La risposta, ovviamente, è positiva, e si incarna nel celeberrimo **grep**, il tool che consente di effettuare ricerche nei file. Il comando

```
grep "Linux Pro" *
```

restituisce tutte le occorrenze della stringa "Linux Pro" (da porre all'interno degli apici, per evitare che le parole successive alla prima siano scambiate da grep per il nome dei file al cui interno operare la ricerca) contenute nei file della directory corrente (**Fig 8**). Come possiamo notare, l'elenco si limita a mostrare la stringa contenuta nel file salve.txt, senza alcuna menzione per il file saluti, che pure fa riferimento al nome della nostra rivista. Come spesso accade sotto Linux, infatti, il comando grep è case sensitive, e pertanto considera come differenti le stringhe "Linux Pro" (salve.txt) e "linux pro" (saluti). Per segnalare a grep la necessità di equiparare le due stringhe, ovvero per optare per una ricerca case insensitive, è possibile ricorrere alla opzione **-i**: il comando

```
grep -i "Linux Pro" *
```

produce l'ouput desiderato. L'aspetto estetico può essere migliorato aggiungendo l'opzione **–color** per evidenziare ciascuna occorrenza:

```
grep –color "Linux Pro" *
```

Se invece si desidera conoscere il nome dei file contenenti la stringa ricercata, è necessario utilizzare l'opzione **-l**; il comando

```
grep -il "Linux Pro" *
```

stampa a video il nome dei file salve.txt e saluti, gli unici contenenti il nome della nostra rivista. Laddove lo desiderassimo, potremmo affinare la nostra ricerca, limitandoci a verificare quali siano le occorrenze del nome della rivista all'interno dei soli file con estensione txt:

```
grep -il "Linux Pro" *.txt
```

stampa a video il solo file salve.txt, l'unico con l'estensione desiderata a contenere riferimenti a Linux Pro. Nei moderni sistemi, così come negli smartphone attuali, la mole di dati immagazzinati è enorme. Ritrovare un'informazione su un hard disk (ma anche su una SD card inserita nel telefono) può rivelarsi un'operazione complessa: se il file ricercato non è di tipo binario, tuttavia, un tool come grep può costituire un prezioso alleato, utile tanto nelle Linux box che negli smartphone. Un altro comando che trova pratico utilizzo tanto sui PC che sui telefoni è **du**, grazie al quale è possibile conoscere la dimensione di un'intera cartella. Per esempio, il comando

```
du .
```

eseguito all'interno della nostra directory di lavoro, consente di verificare come i tre file appena creati abbiano una dimensione congiunta pari ad appena 256 byte (**Fig 9**). Eseguito su una directory contenente una o più subdirectory, il comando restituisce la dimensione di ciascuna sottodirectory; nel caso in cui invece si desideri di conoscere la dimensione di un intero sottoalbero del filesystem,

```
u0_a92@a50ne:/storage/sdcard0/lxp $ grep "Linux Pro" *
salve.txt:salve, lettori di Linux Pro!
u0_a92@a50ne:/storage/sdcard0/lxp $ |
```

› Fig 8: Se non si specifica l'opzione '-i', la ricerca condotta con grep è case sensitive!

è possibile utilizzare l'opzione **-s**. Per esempio, con i comandi

```
cd /storage/sdcard0
du -s .
```

è possibile scoprire la quantità di spazio attualmente impegnata sulla partizione ***/storage/sdcard0*** (al netto di eventuali sottodirectory non accessibili in considerazione dell'assenza dei privilegi di root per l'utente corrente). Tornando alla nostra directory lxp (con il comando: **cd lxp**), possiamo completare il nostro tour sui comandi base per la gestione del filesystem provvedendo a:

» creare due directory annidate:

```
mkdir dir1
mkdir dir2
```

» copiare in dir1 il file buongiorno.txt, mediante il comando **cp**:

```
cp buongiorno.txt  dir1
```

» spostare il file saluti nella directory dir1, rinominandolo contestualmente in saluti.txt, grazie al ricorso al comando **mv**:

```
mv saluti dir1/saluti.txt
```

» cancellare dir2, con il comando **rmdir** (utilizzabile solo su directory vuote):

```
rmdir dir2
```

» cancellare il file buongiorno.txt, del quale è stata appena fatta una copia, con il comando **rm**:

```
rm buongiorno.txt
```

Gestione dei processi

I comandi visti sinora, sebbene preziosi, costituiscono la base per l'utilizzo della shell. L'emulatore di terminale, tuttavia, offre anche qualche comando più avanzato, specie per quanto riguarda la gestione dei processi. Uno dei maggiori crucci per gli utenti di Android "avanzati" è costituito dall'assenza di strumenti immediati per conoscere, a colpo d'occhio, tutte le informazioni salienti (come per esempio utente proprietario, identificatore, occupazione di memoria e di processore, stato, programma di riferimento) sui processi in esecuzione. Il comando **ps** è pensato proprio per colmare questa lacuna: digitando

```
ps
```

(**ps aux** sotto Linux) si può ottenere, dopo la pressione del tasto **Invio**, l'elenco di tutti i processi attualmente in esecuzione. Per ciascun processo sono indicati vari parametri, tra cui è interessante notare:

» il nome dell'utente che lo ha avviato (colonna user);

» l'identificativo univoco o PID, Process ID (colonna PID);

» l'identificativo univoco del processo padre, ovvero del processo che lo ha generato (colonna PPID);

» la quantità di memoria virtuale in carico al processo (colonna VSIZE);

» la dimensione del Residente Set (colonna RSS); »

```
u0_a92@a50ne:/storage/sdcard0/lxp $ du .
256     .
u0_a92@a50ne:/storage/sdcard0/lxp $
```

› Fig 9: Ed ecco un esempio del comando *du*: utilissimo per capire quanto sia striminzito il contenuto della nostra directory di prova

```
u0_a92@a50ne:/storage/sdcard0/lxp $ ps
USER     PID   PPID  VSIZE  RSS     WCHAN    PC         NAME
root      1     0     672    340   ffffffff 00000000 S /init
root      2     0     0      0     ffffffff 00000000 S kthreadd
root      3     2     0      0     ffffffff 00000000 S ksoftirqd/0
root      6     2     0      0     ffffffff 00000000 S migration/0
root      16    2     0      0     ffffffff 00000000 S khelper
root      17    2     0      0     ffffffff 00000000 S fs_sync
root      18    2     0      0     ffffffff 00000000 S suspend
root      19    2     0      0     ffffffff 00000000 S sync_supers
root      20    2     0      0     ffffffff 00000000 S bdi-default
```

› Un estratto del lungo output di ps sullo smartphone in nostro possesso

» lo stato e il nome del programma la cui esecuzione ha determinato la nascita del processo (colonna name). Si tratta di una lista piuttosto lunga e di difficile lettura: per semplificare l'approccio al comando, è opportuno innanzitutto limitarne la verbosità "tagliando" l'output in modo da visualizzare i dati di un singolo processo. A tal fine possiamo ricorrere all'operatore di pipeline, **|**, in grado di redirigere l'output di un comando nell'input di un secondo. Il comando

```
ps | grep /bin/sh
```

consente di effettuare il filtraggio, attraverso grep, dei risultati di **ps**: il lunghissimo elenco prodotto in output da ps viene inviato a grep, che si occupa di visualizzare la sola riga contenente la stringa **/bin/sh**, ovvero i dati del processo relativo alla shell corrente (**Fig 10**). L'output fornito da ps risulta molto utile in diverse circostanze, come vedremo nel proseguo, ma non riesce, tuttavia, a dare un'idea dell'evoluzione dei processi in esecuzione sul sistema. Un processo non è assimilabile a un'entità statica, ma al contrario durante la propria vita "evolve" attraverso una serie di stati, modificando il suo impatto sulle risorse del sistema in ragione di questi, del comportamento previsto dal programma che lo controlla e, per i processi interattivi, degli input dell'utente. In altri termini, fattori come l'utilizzo del processore e l'occupazione di memoria variano dinamicamente durante la vita di ciascun processo, e pertanto un output puntuale come quello di ps (che "fotografa" lo stato dei processi in un determinato istante di tempo) non si adatta a descriverne l'evoluzione in maniera esauriente. Per ovviare a questa limitazione, nei sistemi GNU/Linux si ricorre al comando **top**, supportato anche dal nostro emulatore di terminale e in grado di mostrare l'evoluzione nel tempo (con una periodicità a scelta dall'utente) dei "parametri vitali" che caratterizzano ciascun processo. Strumento versatile, top

Tempi di accesso ai file

In ambiente GNU/Linux, a ciascun file sono associati 3 tempi:

» **Modification time**, mostrato da **ls** con l'opzione **-l**, che indica l'istante in cui è avvenuta l'ultima modifica ai dati contenuti nel file;

» **Change time**, che segnala l'istante in cui è stata effettuata l'ultima modifica alle proprietà del file (come permessi d'accesso, proprietario, ecc.);

» **Access time**, in cui il sistema memorizza l'istante dell'ultimo accesso al file.

```
u0_a92@a50ne:/storage/sdcard0/lxp $ ps | grep bin/sh
u0_a92    17198 17172 996    500   c001169c 4003f5a8 S /system/bin/sh

u0_a92@a50ne:/storage/sdcard0/lxp $
```

› Fig 10: L'utilizzo congiunto di *ps* e *grep* consente di trovare con semplicità i dati relativi ad un processo arbitrario

consente un elevato grado di configurabilità, permettendo all'utente di scegliere, a seconda delle proprie esigenze, il criterio di ordinamento da utilizzare per l'elenco dei processi: come visto con ps, infatti, tale elenco può essere decisamente numeroso, rendendone particolarmente difficoltosa la lettura. Per esempio, il comando

```
top -m 10 -s cpu -d 5
```

comporta, nel nostro emulatore di terminale (sotto Linux la sintassi differisce leggermente, ma i concetti sono i medesimi) la visualizzazione dei primi 10 processi (-m 10), tra tutti quelli in esecuzione, ordinati per percentuale di utilizzo del processore (-s cpu). Lo stato dei processi è aggiornato ogni 5 secondi (-d 5), al fine di consentirci di visualizzare correttamente l'output di top senza quel fastidioso effetto di scrolling dovuto al refresh dei dati. Se manteniamo attivo questo comando per circa un minuto, possiamo notare come la situazione iniziale (**Fig 11**) tenda a evolversi con il passare del tempo, determinando una modifica dell'ordine dei processi. Siete curiosi di conoscere quale sia il processo che rallenta all'improvviso il vostro notebook? Top può fornirvi tutte le risposte che cercate, aiutandovi a individuare anomale percentuali di utilizzo del processore da parte di uno o più programmi. Vale lo stesso discorso per lo smartphone, ove top si rivela ancor più prezioso, in considerazione dell'assenza di un'applicazione nativa in grado di fornire con la medesima immediatezza la risposta al quesito. Spesso tuttavia il problema dei rallentamenti non è, tanto sui PC che sui telefoni, da imputarsi all'utilizzo massivo della CPU, quanto piuttosto all'eccessiva occupazione di memoria volatile da parte dei processi in esecuzione: in questo caso è più utile avviare top con l'opzione **-s rss**, per ordinare i processi in base alla quantità di memoria attualmente in uso. Per uscire dal programma occorre utilizzare il tasto **q** oppure la combinazione **CTRL + C**, viceversa top continuerà a mostrare all'infinito lo stato dei processi monitorati. Proprio questa caratteristica ci può essere utile per fornire una dimostrazione su campo dell'ultimo comando di questa panoramica, il comando **kill**. Apriamo una seconda finestra all'interno dell'emulatore, mediante un tocco sul pulsante **+** visualizzato sull'angolo

```
u0_a92@a50ne:/ $ top -m 10 -s cpu -d 5

User 2%, System 2%, IOW 0%, IRQ 0%

User 11 + Nice 1 + Sys 14 + Idle 476 + IOW 0 + IRQ 0 + SIRQ 0 = 502

  PID PR CPU% S  #THR     VSS     RSS PCY UID      Name
 7501  0   1% R     1   1380K    372K  fg u0_a92   top
  563  0   1% S    92 597182K  39610K  fg system   system_server
23580  0   0% S    52 538436K  25780K  fg radio    com.android.phone
  657  0   0% S    50 561530K  33956K  fg u0_a71   com.android.systemui
  129  0   0% S    45 117224K   3104K  fg system   /system/bin/surfaceflinger
 6299  0   0% S     1      0K      0K     root     kworker/0:2
```

› Fig 11: La situazione iniziale mostrata da *top*, pronta a cambiare al prossimo refresh

```
u0_a92@a58re:/ $ ps | grep top
u0_a92    3977  3924  1404   604   c0063804 400453c8 S top
u0_a92@a58re:/ $
```

› **Fig 12 I dati del processo originato all'avvio del programma top. Si noti, in particolare, il PID attribuito al processo**

in alto a destra dell'interfaccia: possiamo passare in ogni momento da questa alla finestra su cui abbiamo lavorato sinora, ricorrendo al menu a tendina visualizzato nell'angolo in alto a sinistra. Utilizziamo la seconda finestra per ottenere informazioni sul processo generato dall'esecuzione di top: al pari degli altri, sarà caratterizzato da un PID univoco, che costituisce proprio l'argomento richiesto da kill. Come sappiamo, il comando **ps** può fornirci agevolmente questa informazione: è sufficiente reiterare il procedimento già visto, e ricorrere a una pipe per filtrare attraverso grep i dati di top. Il comando

```
ps | grep top
```

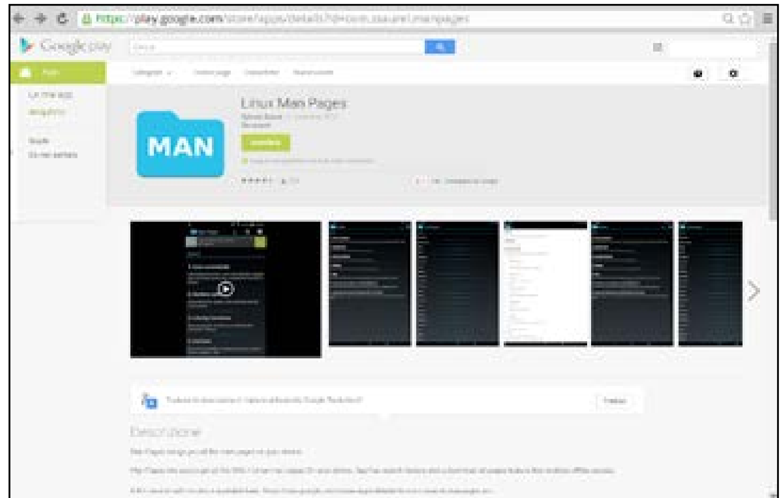
restituisce la riga dell'output di ps relativa al processo originato da top (**Fig 12**), contenente l'indicazione, nella seconda colonna, del PID desiderato. È quanto basta per mostrare un utilizzo pratico del comando **kill**, in grado di "uccidere" un processo, determinandone la chiusura prematura, a partire dal suo PID. Con

```
kill 3977
```

viene provocata la chiusura di top, come possiamo verificare tornando alla finestra numero 1 dell'emulatore.

Installazione del manuale

La maggior parte dei comandi visti sin qui è contraddistinta da un set di argomenti particolarmente numeroso, non sviscerabile, per ovvi motivi di spazio, in questa sede. Appresi i rudimenti di ciascun comando, non resta che approfondirne le finezze attraverso l'esercizio e lo studio personale. Come? Ricorrendo al naturale compagno di ogni utente di shell: il manuale. Android offre molteplici app in grado di visualizzare il contenuto del manuale presente in ambiente Linux (e qui consultabile direttamente da shell, mediante il comando **man**): tra queste possiamo citare **Man Pages** (**Fig 13**) che, al contrario di altre applicazioni, non richiede necessariamente una connessione a Internet per la visualizzazione delle informazioni sui comandi. L'app dispone infatti di un apposito pulsante per il download dei dati relativi ai comandi supportati: in meno di 10 MB aggiuntivi è possibile disporre delle pagine di manuale relative ai comandi più comuni nonché, per i programmatori, alle chiamate di sistema più utilizzate. La disponibilità del manuale durante le sessioni di pratica



› **Fig 13: La pagina del manuale su Google Play, da cui è possibile installare l'app gratuitamente**

(incluse quelle condotte sotto Android) costituisce un valore aggiunto quasi irrinunciabile, se si vuole ottenere in breve tempo la piena padronanza delle opzioni più comuni. Ciò è particolarmente vero per i comandi contenuti nel set aggiuntivo che sarà installato nella prossima puntata, i quali al contrario di quelli visti in questa sede ricalcano con maggiore fedeltà la sintassi e le opzioni presenti in una qualsiasi distribuzione GNU/Linux.

Tirando le somme...

I comandi illustrati in questa puntata rappresentano, pur con le limitazioni dovute all'implementazione minimale offerta dall'emulatore di terminale (limitazioni che, come anticipato, saranno superate nella prossima puntata, con l'installazione del set di comandi aggiuntivi), un utilissimo strumento per apprendere le basi della shell nel campo dell'amministrazione base del filesystem e della gestione dei processi. In aggiunta a questo aspetto, la disponibilità di alcuni dei comandi esaminati consente, in ambiente Android, di sostituire app molto più pesanti sotto il profilo dell'occupazione di memoria e dello spazio su disco, mantenendo inalterato il livello di funzionalità desiderato: si pensi alle applicazioni per la ricerca di contenuti non binari nei file immagazzinati su card SD, egregiamente sostituibili da grep, dei tool per la visione dei processi attivi e/o per il monitoraggio del loro consumo di risorse, o ancora alle applicazioni per la navigazione del filesystem, le cui funzioni principali possono essere svolte dai comandi **mkdir**, **cp**, **cd**, **mv** e **rm**... LXP

Variabili d'ambiente

All'interno della shell è possibile per l'utente definire apposite variabili per conservare nel tempo informazioni d'interesse. Alcune variabili, tuttavia, sono predefinite dal sistema, al fine di preservare informazioni d'interesse generale. Tali variabili sono caratterizzate, oltre che dal prefisso $ tipico di ogni variabile di shell, dal nome costituito interamente da lettere maiuscole, come appunto $SHELL.

Ecco un elenco di alcune delle variabili d'ambiente disponibili in ambiente GNU/Linux, supportate anche dal nostro emulatore di terminale sotto Android (e quindi visualizzabili con il comando echo):

» **$HOSTNAME**, contenente il nome della macchina;
» **$PWD**, che immagazzina la directory di lavoro corrente;
» **$HOME**, ove è indicata la home directory dell'utente;
» **$USER**, contenente il nome dell'utente corrente;
» **$PATH**, che indica le directory al cui interno la shell è tenuta a ricercare gli eseguibili richiesti dall'utente;
» **$PS1**, che fissa il formato del prompt mostrato dalla shell.

Creare giochi in HTML5

Ecco come usare il motore fisico JavaScript **Matter.js** per creare un semplice ed entusiasmante gioco Web in HTML5



Matter.js è un motore di corpo rigido 2D flessibile che si concentra sulle elevate performance, stabilità, facilità d'uso e compatibilità cross-platform. Anche se non è pensato come motore di gioco completo, più come componente, include molte delle cose di cui avete bisogno per cominciare, incluso un ciclo game, controller per mouse, sistema di collisioni e un renderer. Il progetto è attualmente in alpha e ha i soliti avvisi (per esempio, non usare in produzioni critiche), tuttavia è una buona scelta per giochi 2D indie che non richiedono un motore complesso. Potreste obiettare che state già usando qualche altro motore: nessun problema, dal momento che Matter.js include un sistema di callback per gli eventi e controller configurabili, dovrebbe essere possibile integrare il motore in qualsiasi sistema JavaScript o libreria di rendering che vogliate usare. Il motore include un renderer di esempio basato su canvas, con supporto basilare per il rendering delle primitive (vettori) e sprite (texture) dei corpi fisici del mondo. Se doveste aver bisogno di altro, è facile da usare come base per il vostro personale renderer. Se canvas è un po' troppo pedestre per i vostri gusti e preferite la potenza di WebGL, siete fortunati: sostituite nell'esempio incluso il renderer WebGL (che sfrutta la superba libreria Pixi.js) e siete pronti a partire. È possibile anche estendere questo renderer per usare qualcosa come three.js. Nonostante sia una piccola libreria (meno di 200 KB o 53 KB minificata). Matter.js riesce a implementare molte tecniche complicate della fisica dei giochi, e lo fa in un modo molto più leggibile di molti progetti più grossi. Nello specifico, la libreria comprende:

- Posizione corretta in base al tempo con integrale di Verlet;
- Rilevamento adattivo delle collisioni broad-phase;
- Rilevamento AABB mid-phase;
- Rilevamento SAT narrow-phase;
- Risolutore di impulsi sequenziali iterativi e risolutore di posizione;
- Collisioni statiche con limiti statici simili al metodo di Erin Catto;
- Caching e warming per impulsi di coerenza temporale;
- Coppie di collisioni, contatti e impulsi mantenuti con gestore di coppie;
- Modello approssimato di frizione di Coulomb utilizzando limiti di frizione;
- Limiti risolti con il metodo di Gauss-Siedel;
- Timestepping semivariabile sincronizzato con il rendering;
- Una strategia di sleeping basilare;
- Renderer HTML5 Canvas/WebGL.



› ***Matter.js* è un motore altamente scalabile, il che permette di gestire molti corpi rigidi simulati**

Tip

Potete scoprire molto altro su Matter.js alla pagina del progetto **http://brm.io/matter-js**, dove potrete trovare anche diversi demo e una documentazione completa.

Creare un semplice gioco

Ecco come cominciare a mettere le mani in pasta con Matter.js. Costruirete una semplice scena: una piramide di blocchi, una fionda e un masso da lanciare e con cui distruggere la piramide. Se in un qualsiasi momento vi sentite persi, o se qualcosa non vi torna, potete sempre controllare gli esempi e la documentazione per le API sulla pagina del progetto (**http://brm.io/matter-js**). Per prima cosa recuperate una copia del motore dal disco o dall'homepage del progetto e create una pagina HTML vuota impostata con il riferimento allo script:

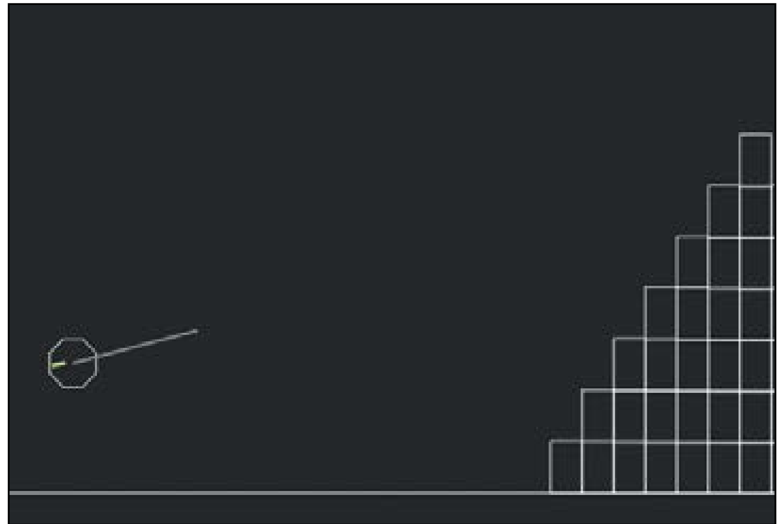
```
<html>
<script src="matter-0.8.0.js" type="text/javascript">
</script>
```

Potete anche caricare la versione minificata, se volete: per questo tutorial entrambe andranno bene. Aprite un altro blocco script pronto per il codice sottostante. Praticamente tutto il codice rimanente andrà in una funzione chiamata **init** che dichiarerete così:

```
<script>
// Matter.js - http://brm.io/matter-js/
function init() {
```

È stato omesso un livello di indentazione nel resto del codice per aiutare la leggibilità: non dimenticate che fa sempre parte della funzione **init()**. Quando includete Matter.js, l'oggetto **Matter**, che contiene tutte le funzioni di Matter, diventa disponibile globalmente. Per mantenere le cose leggibili, è buona idea fare un alias di tutti i moduli che userete all'inizio dello script, così:

```
var Engine = Matter.Engine,
        World = Matter.World,
        Bodies = Matter.Bodies,
        Constraint = Matter.Constraint,
        Composites = Matter.Composites,
        MouseConstraint = Matter.MouseConstraint,
        Events = Matter.Events;
```



› Come semplice demo, questa è la fionda creata in questo tutorial

Per prima cosa create un motore, passando l'elemento per l'inserimento del canvas:

```
var engine = Engine.create(document.body);
```

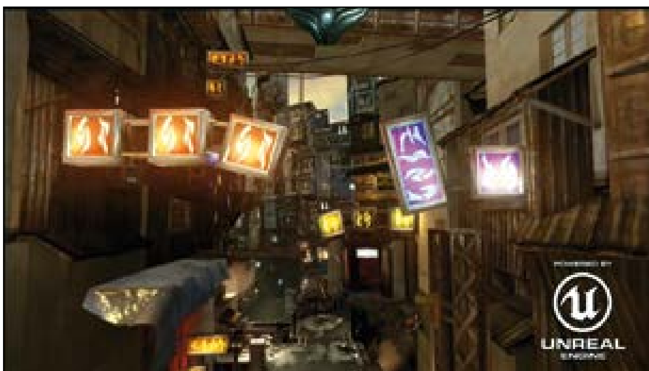
Volete che la scena sia controllabile con il mouse e il motore include un comodo constraint chiamato **MouseConstraint**, quindi aggiungetelo:

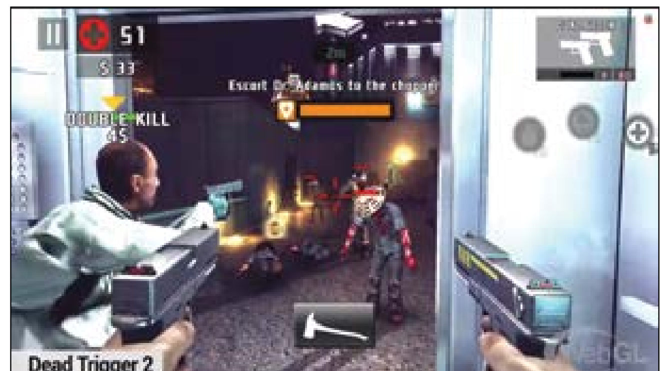
```
var mouse = MouseConstraint.create(engine, { constraint: {
stiffness: 1 }
});
```

Qui avete passato il motore, richiesto da MouseConstraint per funzionare; in più è stato passato un oggetto opzione che modifica la **stiffness** (rigidità) per renderlo meno fiacco »

Perché tanti giochi "fisici" in HTML5?

Come avete visto dal successo dei giochi basati sulla fisica negli app store mobile, la fisica è spesso un componente principale dei giochi moderni e molti la sfruttano come meccanica principale. Perché? Sembra che ci sia una certa soddisfazione nel creare e distruggere cose! Qualche anno fa l'idea di creare giochi reali su tecnologia basata sul Web sarebbe sembrata folle. Fortunatamente la tecnologia Web sta diventando rapidamente un ottimo modo per rendere disponibili i vostri contenuti in un vasto numero di piattaforme, senza (molte) modifiche. Recentemente ci sono stati massicci avanzamenti nella tecnologia HTML5, dalle performance dei motori JavaScript (V8, SpiderMonkey e via dicendo) ad API come canvas e WebGL. Ed è destinato solamente a migliorare. Recentemente Mozilla ha lavorato sia con **Unity** (**netm.ag/unitywebgl-255**) e **Unreal Engine** (**netm.ag/unrealwebgl-255**) per renderli funzionanti a una velocità vicina al nativo nel browser, con l'aiuto di asm.js, una versione di JavaScript altamente ottimizzata. Matter.js potrebbe non essere in tale campionato, non è un motore per il gaming così sviluppato come Unity e Unreal, ma può essere ugualmente un'ottima scelta per giochi 2D indie che non richiedono un motore così pesante.





› Il popolare engine di giochi *Unity* aggiungerà l'export *WebGL* nella versione 5. L'engine *Unreal* ha già delle demo online, come *Tappy Chicken*

dell'impostazione predefinita. Aggiungete alcuni corpi alla scena. Per prima cosa, avete bisogno di un oggetto terreno, altrimenti i corpi cadranno nel vuoto!

```
var ground = Bodies.rectangle(395, 600, 815, 50, { isStatic: true });
```

Il modulo **Bodies** è una factory che può creare tipi comuni di oggetti geometrici: quadrati, rettangoli, cerchi, triangoli e altri poligoni. Come parametri prende posizione e dimensioni richieste. Prende anche un oggetto opzione che in questo caso permette di definire il terreno come **isStatic** (dal momento che normalmente non si muove).

Creare oggetti fisici

Ora, create una roccia. Sentendovi avventurosi userete un esagono, con un raggio di 20px:

```
var rock = Bodies.polygon(170, 450, 8, 20);
```

Per lanciare il vostro masso vi serve un elastico. Per questo userete un **Constraint**:

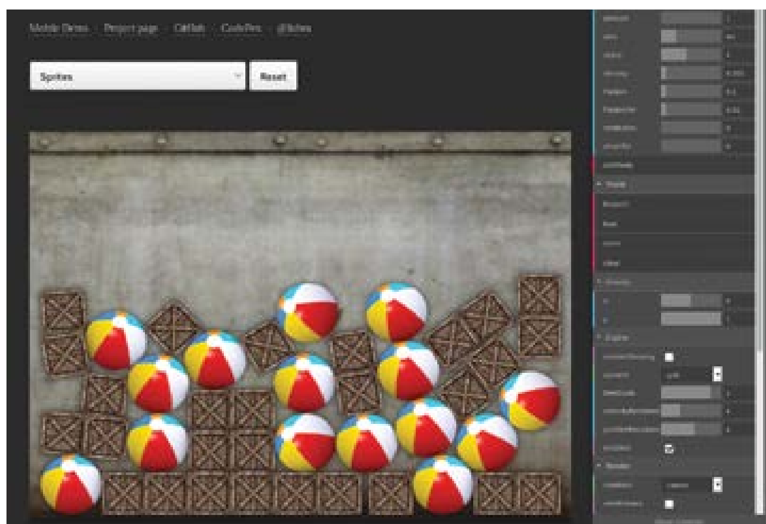
```
var anchor = { x: 170, y: 450 }
elastic = Constraint.create({ pointA: anchor, bodyB: rock, stiffness: 0.1 });
```

Come potete vedere, un capo del **Constraint** è fissato a **pointA**, un'ancora in una posizione prefissata nella scena. L'altro capo è fissato al centro del masso appena creato. La rigidità è stata impostata piuttosto bassa per rendere il Constraint più simile a un elastico, così da dare un po' di potenza alla fionda. La prossima cosa che vi serve è un bersaglio da distruggere. Il motore supporta corpi compositi, ovvero collezioni di **Body**, **Constraint** e anche ulteriori corpi compositi che possono essere manipolati come gruppo. Quello che vi serve qui è una piramide di blocchi, che normalmente richiede un po' di codice per essere impostata. Tuttavia, dal momento che è una richiesta comune, Matter.js include una factory **Composite** che può creare facilmente pile e piramidi:

```
var pyramid = Composites.pyramid(450, 300, 13, 10, 0, 0, function(x, y, column, row) {
            return Bodies.rectangle(x, y, 25, 40);
 });
```

Non ci si addentrerà nei dettagli in questa sede, ma potete trovarli nella documentazione delle API nella pagina del progetto (essenzialmente la factory per la piramide sfrutta una callback nella quale create i vostri corpi e la factory li impilerà correttamente al posto vostro). Ora avete tutti i corpi e anche i constraint, quindi aggiungeteli al mondo:

```
World.add(engine.world, [mouse, ground, pyramid, rock, elastic]);
```



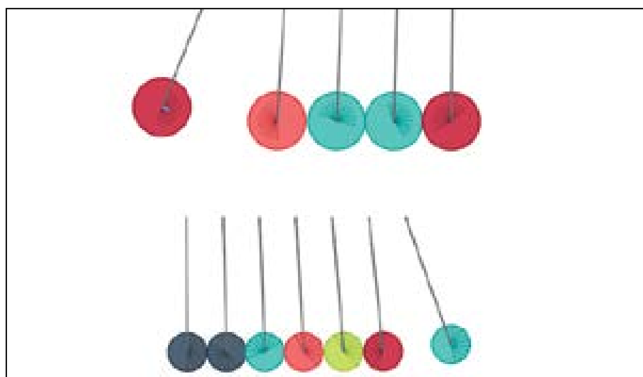
› La demo *Sprites* mostra che non dev'essere tutto tinta unita e mesh

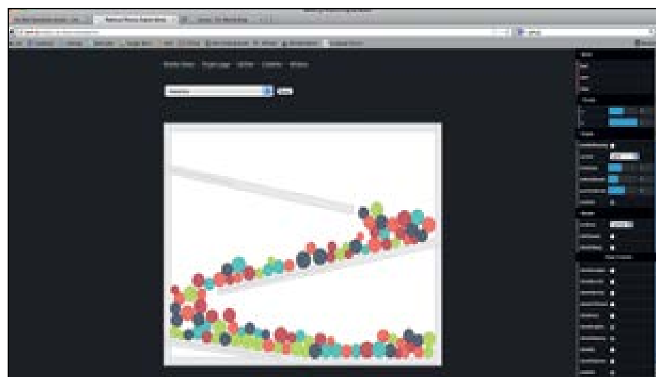
Impostare la logica di gioco

La parte finale è l'aggiunta della logica di gioco. Avete bisogno di rilasciare l'oggetto roccia dall'elastico quando il giocatore lo lascia andare dalla fionda, quindi ricaricare quest'ultima con un altro masso. Per fare questo legherete un evento al motore. L'evento tick viene chiamato a ogni

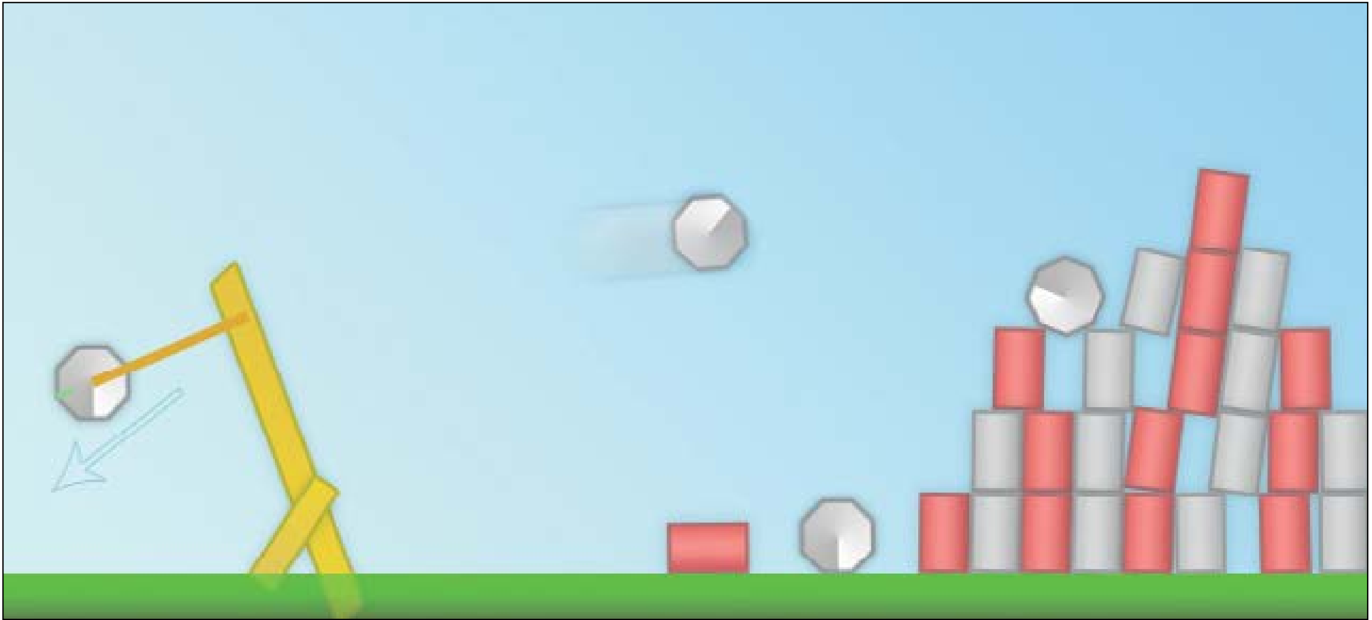
Esempio di un gioco

Potete già vedere 20 diversi esempi e demo nella pagina di progetto di Matter.js (**http://brm.io/matter-js**), incluse le classiche sfere di Newton (**sotto**) che, oltre a essere leggermente ipnotiche, dimostrano le proprietà di conservazione del momento dell'engine. C'è anche una demo specifica per mobile, che include codice per spostare la gravità in base all'accelerometro del dispositivo. Anche se al momento è fluido solamente in dispositivi ben carrozzati, mostra il potenziale per l'uso mobile, anche a questo stadio embrionale. Le demo nella pagina del progetto usano grafiche semplici bianco-su-nero, ma potete trovare delle versioni più colorate della demo gravity su CodePen (**http://codepen.io/liabru**) o visitare la demo completa online (**http://brm.io/matter-js-demo**) e giocare voi con le impostazioni di rendering.





› Volete esplorare i parametri a disposizione? Provate gli effetti delle impostazioni di Matter.js nella demo completo alla pagina del progetto



aggiornamento del motore. Questo significa normalmente che ci sono 60 tick al secondo (a 60fps):

```
Events.on(engine, 'tick', function(event) {
        if (engine.input.mouse.button === -1 && rock.
position.x > 190) {
                rock = Bodies.polygon(170, 450, 7, 20)
                World.add(engine.world, rock);
                elastic.bodyB = rock;
        }
});
```

Questo codice fa diverse cose. Per primo, controlla che nessun pulsante del mouse sia attualmente premuto; se così è, controlla che il masso si sia mosso orizzontalmente oltre la sua posizione iniziale mentre era sull'elastico. Se è vero, il masso dev'essere rilasciato. Il codice quindi crea un nuovo masso e lo attacca all'elastico, questo implica che quello precedente non è più sull'elastico ed è stato rilasciato, sperando che stia volando verso la piramide di blocchi. Per di più, ora avete un nuovo masso nella fionda pronto a partire. Ora che avete impostato tutto correttamente, l'ultima cosa da fare è lanciare la simulazione e chiudere l'iniziale funzione **init()**:

```
Engine.run(engine);
}
```

È molto importante assicurarsi che dopo la dichiarazione della funzione **init** la leghiate all'evento window.load, altrimenti non vedrete alcunché. Aggiungete quindi, dopo la vostra init:

```
window.addEventListener('load', init);
```

Non dimenticate di aggiungere le tag **</script>** e **</html>**. Se tutto è corretto, aprendo la pagina in un browser il gioco dovrebbe essere pronto per il test. In caso non dovesse funzionare, controllate nella console se ci sono errori, quindi controllate il collegamento al codice del tutorial. LXP

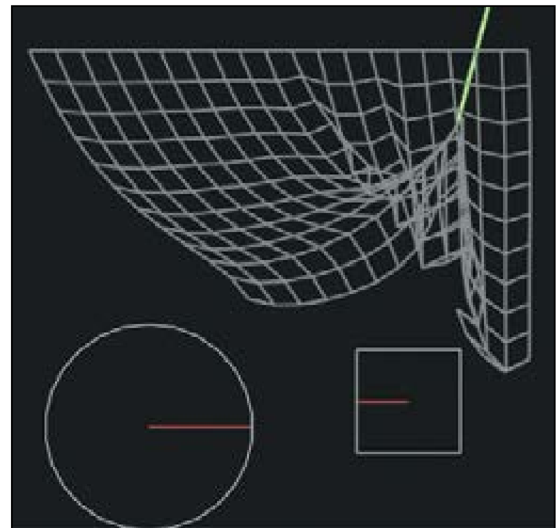
› Anche se non è esattamente Angry Birds, è comunque piuttosto soddisfacente ridurre le piramidi a un cumulo di macerie. Matter.js usa delle tecniche di simulazione per corpi rigidi analoghe al motore Box2D usato nel titolo Rovio, oltre a molti altri

Questione di tele

Questo esempio di tela è uno tra i molti che trovate nelle pagine demo. Potete trascinare i punti nella griglia della tela e cadrà secondo l'effetto della gravità e secondo l'impatto con il cerchio e il quadrato. La tela consiste di 19x11 celle elastiche (risultato dal passaggio di 20 righe e 12 colonne alla classe softBody) che assieme simulano un corpo morbido. L'oggetto **Composites** in Matter.js astrae tutta la complessità nella simulazione, quindi, escludendo tutta la parte di setup (che potete trovare parzialmente prima nel tutorial e il resto nel sito Web), il cuore del codice è piuttosto lineare:

```
var groupId = Body.nextGroupId(),
   particleOptions = { friction: 0.00001,
groupId: groupId, render: { visible:
false }},
   cloth = Composites.softBody(200,
200, 20, 12, 5, 5, false, 8,
particleOptions);
   for (var i = 0; i < 20; i++) {
          cloth.bodies[i].isStatic =
true;
   }
   World.add(world, [ cloth,
          Bodies.circle(300, 500, 80, {
isStatic: true }),
          Bodies.rectangle(500, 480,
80, 80, { isStatic: true })
   ]);
```

L'interattività con il mouse è gestita dal modulo **MouseConstraint** che può gestire anche l'input touch. Matter.js supporta anche altre funzioni specifiche per mobile, come l'input da accelerometri.



Erlang: una guida pratica

Volete cominciare a programmare in **Erlang**? Per prima cosa dimenticate tutto quello che sapete: qui si parla di Programmazione Funzionale!



Erlang è un linguaggio di Programmazione Funzionale sviluppato all'Ericsson Computer Science Labs per lo switching delle apparecchiature telefoniche. Anche se questo lo fa sembrare un linguaggio specifico, non lo è. La Programmazione Funzionale, tuttavia, è un paradigma di programmazione diverso: dovrete dimenticare molte cose che già sapete riguardo la programmazione Procedurale o Orientata agli Oggetti, e cominciare a pensare in un modo diverso. Questo non si applica solamente a Erlang, comunque: altri linguaggi funzionali sono Elixir, SML, Scala e Idris. Il design e la programmazione di Erlang seguono alcuni principi basilari. Il primo è che il sistema fallisce; non potete evitare che prima o poi accada, quindi lasciatelo fallire! Quello che potete evitare è fallire nell'individuazione del fallimento, in altre parole dovete essere in grado di individuare il fallimento. Un altro principio è che le computazioni dovrebbero essere indipendenti per poter usare molteplici CPU e core senza preoccuparvi di dati e memoria condivisi. Questo porta a un simpatico effetto collaterale: potete scalare meglio! (Il sito **www.erlang.org** descrive Erlang come 'massicciamente scalabile'). I processi Erlang parlano tra di loro via messaggi ed Erlang offre un ottimo meccanismo di comunicazione. Erlang può anche essere usato per Web service con l'aiuto della libreria **MochiWeb**.

Numeri e Variabili

Le variabili in Erlang iniziano con Maiuscole, mentre le 'parole' che iniziano con minuscola sono chiamate Atomi che sono costanti con il proprio nome e valore. Dopo aver dichiarato una variabile all'interno della shell Erlang, non potete modificarne il valore:

```
1> MyVar = 12.
12
2> MyVar = MyVar + 5.
** exception error: no match of right hand side value 17
3> MyVar = 5.
** exception error: no match of right hand side value 5
```

Erlang usa il modello ad assegnamento singolo, che permette di assegnare un valore a ogni variabile solamente una volta per sessione o contesto; pertanto, Erlang interpreta "MyVar = MyVar + 5" come "12 = 12 + 5" che da un punto di vista matematico è ovviamente un errore. Il codice Erlang non

OTP

OTP sta per *Open Telecom Platform* ed è la collezione di Erlang di librerie e strumenti Open. Nonostante il nome, OTP riguarda ben più che telefoni! OTP fu disegnata per la creazione di progetti grossi con altrettanto grossi team. OTP riguarda la componentizzazione di tutti i componenti generici e la loro integrazione in librerie riusabili il più possibile. Al programmatore rimane quindi solamente da gestire ciò che cambia da applicazione ad applicazione, mentre OTP si occupa di tutte le cose standard. Se state per scrivere del software in Erlang, quindi, dovrete far uso del framework OTP che vi permette di gestire il codice esistente. Per gestire un modulo esistente, dovrete scriverne un altro ma senza effettuare alcuna modifica al primo! Una volta che il supervisor identifica che il processo server non è in esecuzione, automaticamente lo riavvia. OTP è la parte più importante e vitale di Erlang. Una visione più dettagliata di Erlang merita un articolo futuro a sé: tenete gli occhi aperti!

assomiglia a codice nel senso tradizionale (C) del termine. Il modo di programmare in Erlang si concentra più sul flusso di dati, ciò significa che dovrete imparare a disegnare i vostri programmi basandovi sulla ricorsione e sul passaggio di messaggi. L'editor integrato nella shell di Erlang (a.k.a. *erl*) è un dialetto di *Emacs*. La shell di Erlang è il posto perfetto per controllare la maggior parte del vostro codice prima di creare un modulo. Prima di continuare il resto del tutorial, imparate che potete chiudere la shell Erlang scrivendo **q().** e premendo **Invio**.
Il punto alla fine è indispensabile. Il codice per il programma "Hello world" in Erlang, incapsulato in un modulo Erlang è:

```
-module(helloWorld).
-export([hello_world/0]).
hello_world() -> io:fwrite("Hello World!\n").
```

L'istruzione **export([hello_world/0])** indica che la funzione **hello_world()** non prende argomenti. Il primo comando dichiara il nome del modulo. Ora, potete lanciarlo all'interno di una shell Erlang come segue (ammesso che **helloWorld.erl** sia nella directory attuale):

```
$ erl
Erlang/OTP 17 [erts-6.1] [source] [64-bit] [async-threads:10]
[kernel-poll:false]

Eshell V6.1  (abort with ^G)
1> c(helloWorld) %% nessun punto alla fine del comando
1> c(helloWorld). %% interpretato come c(helloWorld)
c(helloWorld).
* 2: syntax error before: c
1> c(helloWorld). %% ora è giusto!
{ok,helloWorld} %% la compilazione è andata bene
2> helloWorld:hello_world(). %% lanciate la funzione voluta
Hello World!
ok
```

Ogni comando Erlang che eseguite all'interno della shell deve finire con un punto per essere eseguito. Il comando **c(helloWorld).** compila il codice sorgente trovato nel file **helloWorld.erl**. Il comando **helloWorld:hello_world().** esegue la funzione chiamata **hello_world()** all'interno del modulo **helloWorld**. Quando eseguite **io:fwrite("Hello World!\n")** nella shell, la funzione **fwrite** stampa l'output indicato e la shell Erlan stampa il valore di ritorno, ovvero "ok". Potete anche compilare il file **helloWorld.erl** e lanciare la funzione che volete dalla shell Linux:

```
$ erl -compile helloWorld.erl
$ ls -l
totale 8
-rw-r--r-- 1 ciromattia ciromattia 580 mar 29 21:45 helloWorld.
beam
-rw-r--r-- 1 ciromattia ciromattia  93 mar 29 21:36 helloWorld.
erl
$ erl -noshell -s helloWorld hello_world -s init stop
Hello World!
```

Le funzioni sono principali in Erlang (così come in qualsiasi linguaggio funzionale) e possono essere usate come argomento per altre funzioni. Le funzioni possono anche ritornare altre funzioni! Erlang supporta **Funs**, ovvero oggetti funzionali. Funs vi permette di creare oggetti anonimi e passare la funzione come argomento ad altre funzioni. Altri tipi di strutture dati supportate da Erlang sono le **Tuple**, **Mappe**, **Liste** e **Record**. Erlang non ha un ciclo **for** perché non vi serve: può essere sostituito dal pattern **matching** e funzioni di ordine più elevato. Potete comunque programmare da voi un ciclo **for**. Un'implementazione possibile è questa:

```
for(MAX, MAX, J) -> [J(MAX)];
for(I, MAX, J) -> [J(I)|for(I+1, MAX, J)].
```

Il codice crea una lista che contiene tutti i numeri prodotti dal ciclo for. Dopo aver inserito tale codice nel file helloWorld.erl (ricordate che tutto il codice Erlang dovrebbe essere all'interno di moduli a meno che non usiate la shell Erlan) e esportato la funzione for, potete usarla come segue:

```
helloWorld:for(1,10,fun(I) -> 3 * I end).
[3,6,9,12,15,18,21,24,27,30]
4> helloWorld:for(-1,5,fun(I) -> I * I * I end).
[-1,0,1,8,27,64,125]
```

Come potete leggere, potete gestire l'output del ciclo for dinamicamente modificando la definizione della funzione fun(I), che è un parametro della funzione for, come fareste in un normale ciclo for. Essere in grado di sviluppare le proprie strutture di controllo è un enorme vantaggio perché non siete limitati da un numero prefissato di strutture di controllo, come succede con i linguaggi procedurali e orientati agli oggetti.
Le strutture di controllo create da voi sono anche solitamente più appropriate per il tipo di problema che vi trovate a risolvere e rendono il vostro codice più leggibile.

Scrivere un semplice programma in Erlang

Il codice per trovare gli interi della sequenza di Fibonacci in Erlang è il seguente (**linuxpro.erl**):

```
-module(linuxpro).
-export([fib/1]).

fib(N) when N == 0 -> 0;
fib(N) when N == 1 -> 1;
fib(N) when N == 2 -> 1;
fib(N) when N > 2 -> fib(N-1)+fib(N-2).
```

La dichiarazione **fib/1** indica che la funzione **fib()** richiede un argomento. La funzione **fib(N)** ha molti rami che dipendono dal valore del parametro. Potete lanciare la funzione **fib** dalla shell Erlang come segue:

```
1> c(linuxpro).
{ok,linuxpro}
2> linuxpro:fib(20).
6765
```

In cima al codice la keyword **-module** definisce il nome del modulo, ovvero **linuxpro**. La direttiva **-export** definisce la lista »

› Fig 1: il codice sorgente completo del modulo *processFile.erl*, incluso il codice per produrre la documentazione HTML

Tip

Dovreste usare sempre Erlang? Erlang non può risolvere qualsiasi problema di programmazione in maniera efficiente, il che significa che potreste aver bisogno di un altro linguaggio di programmazione per problemi specifici. Usate sempre lo strumento adatto a seconda del lavoro da fare!

```
code$ erl
Erlang R15B01 (erts-5.9.1) [source] [64-bit] [async-threads:0] [kernel-poll:false]

Eshell V5.9.1  (abort with ^G)
1> cprof:start().
5614
2> processFile:readfile("data").
even
odd
odd
even
odd
odd
File closed.
"12\n23\n43\n32\n3\n21\n"
3> cprof:pause().
5731
4> cprof:analyse(processFile).
{processFile,14,
             [{{processFile,process_line,2},7},
              {{processFile,save_line,2},6},
              {{processFile,readfile,1},1}]}
5> cprof:stop().
5731
6>
```

› Fig 2: Erlang ha i propri strumenti per il profiling. Qui potete vedere il profilatore *cprof* al lavoro su un semplice modulo Erlang

completa delle funzioni visibili al mondo. Il modulo linuxpro esporta una funzione chiamata **fib/1**. Notate che sia il nome sia il numero totale di parametri di una funzione sono predefiniti. Erlang considera funzioni con lo stesso nome ma diverso numero di parametri come funzioni completamente diverse.

Ancora sui moduli Erlang

I moduli in Erlang sono come le classi nei linguaggi orientati agli oggetti, e i processi in Erlang sono come oggetti.
I processi in Erlang non condividono memoria (cosa buona!) e possono comunicare solamente via messaggi. I processi in Erlang non hanno nulla a che fare con processi o thread Unix. Dopo aver compilato un modulo, sia all'interno della shell Erlang o dalla shell Unix, viene generato un nuovo file con lo stesso nome del modulo ma con estensione **.beam**.

```
$ erl -compile linuxpro.erl
$ ls -l linuxpro.*
-rw-r--r-- 1 ciromattia ciromattia 604 mar 29 21:49 linuxpro.
beam
-rw-r--r-- 1 ciromattia ciromattia 154 mar 29 21:48 linuxpro.erl
```

Una parte fondamentale dell'*Erlang Runtime System* (ERTS) è il processore BEAM. BEAM sta per Bogdan's Erlang Abstract Machine, che è una macchina virtuale per interpretare codice BEAM ottimizzato. Quando compilate un programma Erlang, il compilatore lo converte sempre in questo tipo di codice.

Lavorare con i file

Ora è il momento di imparare come leggere e scrivere file in Erlang programmando il modulo **processFile.erl** (**Fig 1**). Il codice leggerà un file di testo passato come argomento e contenente interi, e scriverà tutti gli interi pari trovati nel file in un file separato chiamato 'LXPpari'. La funzione integrata (BIF) **file:open(FileName, [read])** apre un file in lettura. Per aprire un file in scrittura, userete il codice seguente:

```
{ok, FILE} = file:open("output", write)
```

La parte più importante e interessante del modulo è:

```
save_line(Line, File) ->
        TrimLine = re:replace(Line, "(^\\s+)|(\\s+$)", "",
[global,{return,list}]),
        Number = list_to_integer(TrimLine), if (Number rem
2) == 1 ->
                io:format("odd\n"),
                io:format(File, "~s~n", [TrimLine]);
                        true -> io:format("even\n")
                        end.
```

Tip

Nel creare programmi, i programmatori Erlang dividono il codice in moduli. I moduli vi permettono di organizzare, archiviare e condividere il codice in modo semplice ed efficiente e sono il miglior modo per scrivere programmi Erlang.

Erlang non ha un operatore di ritorno. Tutte le istruzioni sequenziali all'interno dello stesso blocco di codice sono separate da virgole e terminano con un punto e virgola. Un'istruzione **if** deve coprire tutte le possibili opzioni, perciò la riga **true -> io:format("pari\n")** è richiesta anche se non fa nulla di utile. L'ordine dei diversi casi **if** ha senso perché viene eseguito solo il primo che si verifica. Viene utilizzata un'espressione regolare per rimuovere tutti gli spazi, incluso il carattere **newline**, dalla variabile **Line**. La variabile **TrimLine** viene usata per scrivere un intero pari nel file di output.
La stringa viene convertita in intero con l'istruzione **Number = list_to_integer(Trimline)**. Il programma può essere compilato ed eseguito all'interno della shell Erlang come segue:

```
33> c(processFile).
{ok,processFile}
34> processFile:readfile("data").
pari
...
dispari
File closed.
"12\n23\n43\n32\n3\n21\n"
```

Non dovete dimenticare di chiudere il file dopo aver concluso le operazioni di scrittura con **file:close(FILE)**. Quindi, è Erlang adatto per la programmazione di sistema? In realtà no, ma potete comunque adattarlo al bisogno.

Profilazione e testing della copertura di codice

Erlang offre tre strumenti per il profiling: **cprof**, **fprof** e **eprof**. Di questi tre, fprof aggiunge un carico significativo al sistema ma è adatto alla profilazione di grossi sistemi in laboratorio o in ambiente simulato. Gli altri due sono adatti alla profilazione di ambienti live dal momento che hanno un impatto inferiore sulle performance (meno del 10 percento). Potete lanciare cprof con **cprof:start().**, metterlo in pausa con **cprof:pause().** e fermarlo con **cprof:stop().**. Dopo aver lanciato cprof dovreste eseguire il programma che volete profilare, quindi metterlo in pausa. La **Fig 2** mostra il risultato della profilazione del modulo **processFile.erl**. L'output mostra che processFile ha fatto in tutto 14 chiamate di funzione: una chiamata a **readfile()**, sei a **save_line()** e sette a **process_line()**. I risultati hanno senso perché **readfile()** viene chiamata una sola volta e **process_line()** tante volte quante **save_line()** più una, richiesta per identificare la fine del file. Trovare le righe di codice che non vengono eseguite è particolarmente utile dal momento che tali linee sono un chiaro segnale di qualcosa che non va nel codice (bug?). Sapere quali funzioni sono chiamate più spesso è altrettanto utile per sapere cosa conviene ottimizzare. Questo potete farlo con l'analizzatore della copertura di codice, fornito altresì da Erlang. Un test di copertura del codice del modulo **processFile.erl** include i passi seguenti:

```
2> cover:start().
{ok,<0.39.0>}
3> cover:compile(processFile).
{ok,processFile}
4> processFile:readfile("data").
...
5> cover:analyse_to_file(processFile).
{ok,"processFile.COVER.out"}{ok,<0.39.0>}
3> cover:compile(processFile).
{ok,processFile}
```

Le sei regole

Erlang è disegnato per seguire queste sei regole:

» **Isolamento:** i processi Erlan sono isolati per design. Ognuno ha il proprio stack e heap e il garbage collector funziona separatamente. I processi non possono leggere la memoria di altri processi e quindi non possono danneggiarli.

» **Concorrenza:** i processi sono concorrenti, per cui in teoria tutti i processi possono girare in parallelo. Questa è un'ottima cosa per computer con processori multi-core poiché i processi possono dividersi tra i core disponibili.

» **Individuazione degli errori** i processi Erlang sono in grado di riconoscere i failure. Potete anche creare un collegamento tra due processi, per cui quando un processo muore per qualche ragione, un altro processo può esserne informato, così da gestire e risolvere il problema.

» **Identificazione dell'errore:** se un processo va in errore, il segnale d'errore contiene dati aggiuntivi forniti dal runtime Erlang per comunicarvi esattamente cos'ha provocato il guasto.

» **Aggiornamento live del codice:** Erlang può essere modificato finché è in esecuzione! Ciò significa che potete aggiornare le applicazioni finché girano.

» **Archivio stabile:** questo non viene fatto da Erlang ma da librerie di terze parti. Potete usare Mnesia, Riak e altre per archiviare dati. Ogni processo può accedere ai dati dal momento che l'archivio è condiviso tra i processi Erlang.

```
4> processFile:readfile("data").
...
5> cover:analyse_to_file(processFile).
{ok,"processFile.COVER.out"}
```

I risultati vengono salvati automaticamente nel file **processFile.COVER.erl** da visualizzare con il vostro editor preferito.

Erlang e Mnesia

Mnesia è un DBMS distribuito molto veloce, scritto in Erlang e parte della distribuzione standard di Erlang e può memorizzare qualsiasi tipo di struttura dati di Erlang. Mnesia può memorizzare le tabelle sia in memoria che su disco, per velocità o persistenza. Per creare un nuovo database Mnesia dovete eseguire la funzione **mnesia:create_schema()**. Dopo aver lanciato tale funzione, Mnesia crea una nuova struttura di directory per memorizzare i file del database. La **Fig 3** mostra un esempio d'utilizzo completo di Mnesia ed Erlang con il codice del modulo **myMnesia.erl**. Il comando **mnesia:create_schema([node()])** inizializza un nuovo database Mnesia al nodo attuale grazie al parametro **node()**. Notate che le righe di una query di selezione possono essere in qualsiasi ordine. La funzione **traverse_table_and_show()** legge tutti i dati da una tabella e li stampa a schermo. La funzione **add_magazine** è implementata come segue:

```
add_magazine(Number, Year, Month, Pages) ->
            Row = #linuxpro{number=Number, year=Year,
month=Month, pages=Pages},
            F = fun() ->
                        mnesia:write(Row)
                  end,
            mnesia:transaction(F).
```

Il primo comando definisce il nuovo record e lo memorizza nella variabile Row. La funzione **transaction()** esegue l'oggetto funzionale F come transazione. Questo è il modo standard per salvare record Mnesia. Dovreste lanciare la funzione **myMnesia:initialize_database_once()** una sola volta. Dovreste quindi lanciare il database Mnesia manualmente con il comando **mnesia:start()**. Come potete vedere in **Fig 3**, l'implementazione attuale non è per la persistenza dei dati, quindi una volta usciti dalla shell Erlang tutti i dati della tabella 'linuxpro' saranno perduti!

Generare documentazione Erlang

Erlang vi permette di includere metadata di documentazione durante lo sviluppo di un modulo. Dovrete sempre documentare il vostro codice per aiutare altri programmatori così come voi stessi. Le righe extra che dovete scrivere nel modulo **processFile** per documentare il codice non modificheranno la funzionalità o la velocità di esecuzione del modulo; Erlang usa queste righe extra per generare output HTML. Per generare i file HTML e inserirli in un directory chiamata 'docs' eseguite il comando seguente in una shell Erlang:

```
edoc:files(["processFile.erl"], [{dir, "docs"}]).
```

La directory 'docs' ha il seguente contenuto:

```
$ ls -l documentation/
total 32
-rw-r--r-- 1 ciromattia ciromattia 40 Mar 28 19:46 edoc-info
-rw-r--r-- 1 ciromattia ciromattia 2109 Mar 28 19:46 erlang.png
-rw-r--r-- 1 ciromattia ciromattia 471 Mar 28 19:46 index.html
-rw-r--r-- 1 ciromattia ciromattia 424 Mar 28 19:46 modules-
frame.html
-rw-r--r-- 1 ciromattia ciromattia 1070 Mar 28 19:46 overview-
summary.html
-rw-r--r-- 1 ciromattia ciromattia  327 Mar 28 19:46 packages-
frame.html
-rw-r--r-- 1 ciromattia ciromattia 2512 Mar 28 19:46
processFile.html
-rw-r--r-- 1 ciromattia ciromattia 895 Mar 28 19:46 stylesheet.
css
```

Ricordate che la documentazione viene prodotta solo per le funzioni nella lista di esportazione anche se scrivete codice documentale (@doc) per altre funzioni. Come disse una volta Steve Jobs, "L'industria sta andando verso il moltiplicare sempre più i core, ma nessuno sa come programmarli. Due: ok, quattro: ehm, otto: non ci pensare nemmeno". Erlang supporta CPU multicore per design! Più programmate in Erlang, più il vostro codice sarà efficiente. Un ottimo libro su Erlang è **Programming Erlang: Software for a Concurrent World** di Joe Armstrong. LXP

› **Fig 3: una piccola interazione con il database Mnesia utilizzando Erlang**

Tip

Per gestire gli errori dovete sfruttare i componenti isolati. Se i componenti sono isolati possono girare in parallelo. Unendo questi due concetti potete concluderne che l'isolamento permette ai processi di essere paralleli e scalabili.

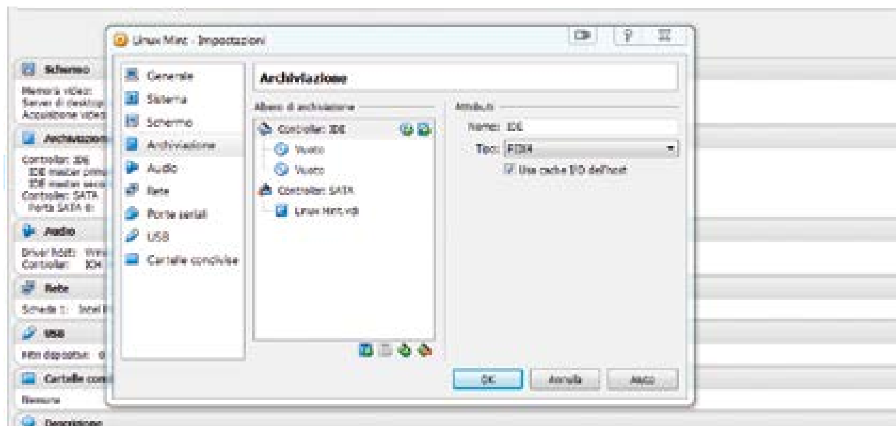
Domande & Risposte

Avete qualche domanda sull'Open Source? Scriveteci a **mailserver@linuxpro.it** per ottenere una risposta

1 Linux Virtualizzato

Utilizzo un sistema operativo Windows 8.1 a 64-bit e vorrei iniziare a usare Linux. Le distribuzioni che mi attirano maggiormente sono Mint e Ubuntu e mi piacerebbe provarle in VirtualBox. Come posso installarle?

Linux Mint o Ubuntu sono due ottime distribuzioni per iniziare. Entrambe hanno un buon grado di flessibilità e in più sono sufficientemente facili da usare anche per chi non ha mai messo mano a una distro Linux. Il procedimento per installarle in VirtualBox, e quindi provarle su una macchina virtuale, è molto semplice ed è valido sia che si stia usando Windows sia un'altra distribuzione. Se per esempio volete provare una distro particolare e non avete intenzione di dedicargli una partizione del disco fisso, installarla su una virtual machine è la soluzione ideale per fare le vostre prove senza pericolo di danneggiare alcunché. Per prima cosa, quindi, procuratevi l'immagine ISO di una delle due distribuzioni. Potete trovarle sui siti ufficiali dei rispettivi progetti. Nel caso di Mint, **www.linuxmint.com**, mentre per Ubuntu, **www.ubuntu-it.org**. Vengono messe a disposizione le versioni a 32 e 64 bit. Se avete un sistema a 64 bit, scegliete quest'ultima, altrimenti optate per la prima. Una volta terminato il download, dovrete procurarvi VirtualBox che potete scaricare dalla pagina **www.virtualbox.org**.
Se state già usando una distro Linux, date un'occhiata nel vostro Software Center. Molto probabilmente troverete il programma di Oracle già pronto per l'installazione, altrimenti, sempre sul sito, sono disponibili svariati pacchetti precompilati per i sistemi più conosciuti, tra cui Ubuntu, Debian, OpenSUSE, SUSE, Fedora, Mandriva, Oracle Linux. Dopo aver installato e avviato VirtualBox, fate click sulla voce **Nuova**, così da iniziare la procedura per la creazione di una macchina virtuale. Qui verrà chiesto di inserire determinate informazioni circa il sistema che volete ospitare, scegliendo poi la quantità di RAM e le dimensioni del disco fisso virtuale che si andranno a generare. Se in fase iniziale inserite il nome della distro da virtualizzare, come Ubuntu o Mint (Debian), VirtualBox vi consiglierà di impostare RAM e spazio dell'hard disk con parametri preconfigurati che potete lasciare invariati. Terminata la creazione della macchina virtuale che avviene tramite questa procedura passo a passo dovrete configurare il sistema perché avvii la ISO della vostra distro. Selezionate con un click del mouse la virtual machine che compare nel menu di sinistra, quindi



› Per aggiungere una ISO a una macchina virtuale in VirtualBox, fate click sull'icona raffigurante un CD con sopra un più (+), di fianco alla voce *Controller: IDE*

Riflettori su...

Perché dopo aver ricompilato il kernel lo spazio diminuisce?

Quando si ricompila il Kernel, di solito lo spazio su disco diminuisce dai 20 ai 30 Megabyte a seconda delle circostanze.
È un calo che potremmo definire fisiologico e quindi perfettamente normale. Infatti, è dovuto alle proprietà intrinseche dell'operazione di ricompilazione che, durante l'esecuzione, tende a occupare più spazio del necessario. Molte persone pensano erroneamente che per risolvere sia necessario cancellare il vecchio kernel, ma non è così. Infatti tutto quello che dovete fare è eseguire un comando che permetta al sistema di riappropriarsi dello spazio indebitamente occupato in fase di ricompilazione. Spostatevi nella directory **/usr/src/linux** e scrivete l'istruzione:

```
make clean
```

Terminale e superutente

Nelle distro Linux ci sono diversi modi per applicare un comando a una data operazione. La maggior parte delle volte, infatti, potete procedere sia utilizzando degli strumenti grafici sia la potenza della riga di comando. In quest'ultimo caso, avrete a disposizione strumenti molto più flessibili e pratici da usare, ma soprattutto validi per tutte le distribuzioni. Prendendo come esempio i comandi per la configurazione di sistema, avrete sicuramente notato come nel momento in cui tentate di eseguirne uno, vi venga richiesto l'accesso come *superuser* o *root*. In Ubuntu e nelle sue derivate, per esempio, questo è identificato con il comando **sudo**, a cui poi deve seguire l'inserimento della password. Altre distro come Fedora, invece, utilizzano **su**, che danno accesso all'utente root fino a quando non si inserisce il comando **logout**.

premete il pulsante **Impostazioni**. Selezionate la funzione **Archiviazione** e fate click sull'icona raffigurante un CD con un più (+) verde di fianco alla voce **Controller IDE**. In questo modo, aggiungete un lettore ottico virtuale. Scegliete adesso l'immagine ISO precedentemente scaricata e confermate con **OK**. A questo punto, non vi resta che avviare la macchina virtuale in questione e seguire la procedura d'installazione della distro scelta.

2 Android e X

Ho un Raspberry Pi a cui sono solito collegarmi tramite SSH con un tablet Nexus 10. Per farlo, utilizzo un'ottima applicazione chiamata JuiceSSH che permette di utilizzare una console di comando. Vorrei sapere se esiste un modo per utilizzare SSH con X tramite il Nexus, così da eseguire applicazioni grafiche sul Raspberry Pi senza bisogno di avviare il desktop LXDE.

Prima di iniziare con una descrizione più dettagliata della procedura da mettere in pratica per eseguire quanto chiesto, è importante fare una premessa. X è un sistema client/server. Ciò significa che i programmi eseguiti sul client vengono poi visualizzati anche sul server. Nell'uso tradizionale, client e server sono sulla stessa macchina ma niente impedisce di separarli come nei casi in cui si tenda a usare un *X forwarding*. In pratica, è necessario un server X sul dispositivo su cui si esegue il *forward*. Android, però, non dispone in modo nativo di alcun supporto per X. Per fortuna esistono diversi server che fanno al caso vostro da usare con questo sistema operativo. Uno di questi è **X Server** di **Darkside Technologies** che, seppure non fornisce un'implementazione completa di X, gestisce abbastanza parametri da soddisfare la maggior parte delle esigenze. In più, è in grado di gestire la procedura di X forwarding con la versione Pro di JuiceSSH. Ipotizziamo quindi di avere un collegamento già attivo con JuiceSSH. Spostatevi nell'elenco delle connessioni e scorrete fino a quando non trovate la lista denominata **Port Forwards**. Qui create un nuovo record utilizzando la connessione per il Pi. Dopo aver scelto il nome, potete utilizzare le seguenti impostazioni:

```
Port (from): 6001
Destination: 127.0.0.1
Port (to): 6000
Open in browser: No
```



› **Ricordate: Android non ha un server X. Per usarlo servono programmi come *X Server* o *XServer SXDL***

Una volta fatto, salvate il tutto. Avviate il server X e poi tornate in JuiceSSH, quindi lanciate la connessione SSH. Avviate il programma in questione con:

```
env DISPLAY=:1 vostraapplicazione
```

Adesso tornate al server X per vedere il software aperto. Non preoccupatevi se si verifica un piccolo ritardo nell'apertura della finestra. È perfettamente normale. Ricordate poi che il numero del DISPLAY è legato al parametro **From** che mostra la porta utilizzata nel vostro profilo. Per esempio, se la porta From è 6002, si dovrebbe utilizzare il parametro **DISPLAY=:2**. Quando avete concluso, tornate in JuiceSSH, quindi disabilitate il port forwarding. X Server è piccolo e leggero ma, come detto in precedenza, non pienamente completo. Se volete un'alternativa molto più robusta, allora date un'occhiata a **XServer SXDL.** Quest'ultimo, infatti, è capace di gestire i programmi che X Server non riesce a supportare. »



Riferimento rapido

dmesg

In ogni discussione dove si parli di problemi di hardware si finisce sempre per fare la stessa domanda: "Cosa riporta **dmesg**?". La prima cosa che si fa è quindi scrivere il comando dmesg in un terminale, finendo per essere sommersi da migliaia di righe di puro codice. Quello che potete vedere davanti ai vostri occhi è un elenco di tutti i messaggi del kernel. Se reindirizzate questo contenuto a un pager, potrete scoprire il modo in cui il vostro hardware di sistema viene rilevato, nonché i corrispondenti driver che vengono caricati. Dmesg produce una quantità eccezionale di informazioni e quindi avrete immediatamente bisogno di raffinare la ricerca, mettendo in risalto ciò che principalmente vi interessa. A questo proposito, ci sono tre opzioni principali che potete usare. La prima è l'invio dell'output a un pager con:

```
dmesg | less
```

L'alternativa è utilizzare grep per filtrare i risultati. Se per esempio state cercando delle informazioni su un dispositivo USB, questa è l'istruzione da cui cominciare:

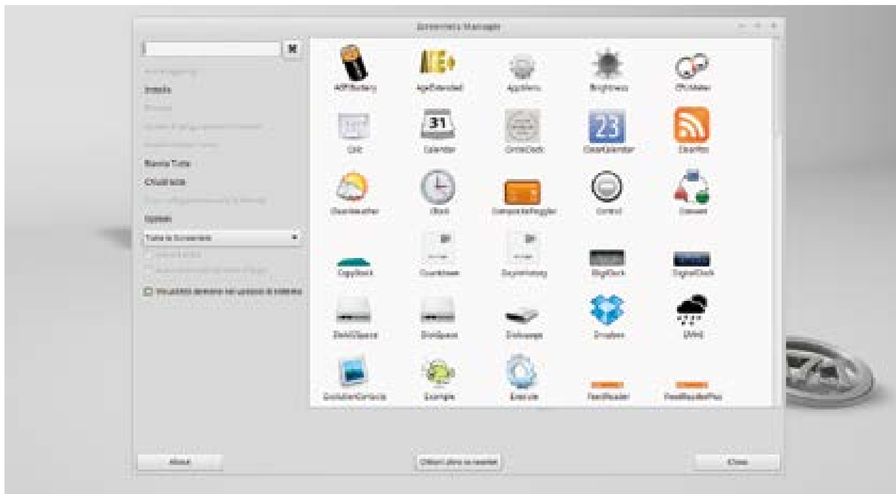
```
dmesg | grep -i USB
```

Potete usare anche le opzioni **-e** o **-T**. In entrambi i casi, queste richiedono a dmesg di utilizzare dei timestamp leggibili, anche se in modi leggermente diversi. La terza opzione è disponibile solo con le versioni più recenti di **util-linux** (il pacchetto di dmesg) – dalla 2.22 in poi. Infatti, è possibile usare l'opzione **-w** o **--follow**. Con **dmesg --follow** vengono prima mostrati i messaggi del kernel, continuando poi a visualizzare quelli nuovi.

» 3 Problemi di screenlets

Ho utilizzato senza problemi gli screenlets su varie edizioni di Linux Mint. Con la versione 17, però, ho riscontrato alcuni malfunzionamenti nello spostamento degli widget ai lati del desktop.

In realtà, non è la prima volta che sentiamo di questi problemi. Gli screenlets sono essenzialmente dei widget che permettono di abbellire il desktop con specifiche funzioni. Allo stato attuale, tuttavia, non esiste una soluzione univoca per risolvere l'inconveniente registrato in Linux Mint 17. L'unica cosa da fare è provare diverse procedure, sperando che una di queste possa fare al caso vostro. Alcuni utenti, per esempio, hanno segnalato che un buon sistema per sbloccare gli screenlets rimasti "incagliati" in una zona del desktop consiste nell'aprire il pannello delle proprietà del componente. Nella scheda **Opzioni**, in primo luogo, è necessario controllare che la voce **Blocca** non sia spuntata. In seconda battuta, agendo sulle funzioni X e Y, è necessario inserire nuovi valori, quindi utilizzare le frecce direzionali per perfezionare la posizione del widget. A questo punto, dovreste vedere lo screenlet muoversi secondo le vostre direttive. Se anche questo sistema fallisce, allora è necessario vederci un po' più chiaro, andando alla ricerca di qualche file di log che spieghi cosa stia succedendo.
Ci sono due sistemi per trovarli. Il primo è dare un'occhiata in **~/.config/screenlets**, mentre il secondo consiste nell'eseguire **screenlets-manager** da una finestra di terminale. Dovrete poi fare attenzione ai vari output, soprattutto quando si tenta di spostare un widget sui bordi del desktop. Infine, se non doveste trovare alcuna soluzione, prendete in considerazione che si tratti di un bug riferito solo ad alcuni screenlets. In questo caso, spesso si può risolvere rimuovendo l'oggetto, eseguendo un aggiornamento di sistema e installando di nuovo il widget in questione.



› Agendo sulle proprietà degli Screenlets si possono risolvere problemi di varia natura

La soluzione del mese

12 GB andati persi!

D Ho fatto un enorme danno. Involontariamente ho finito per cancellare ben 12 GB di dati dal mio disco fisso. Ho dato un'occhiata ai principali forum su Linux Mint, la distro che utilizzo, ma non ho trovato nessun suggerimento in particolare, se non usare *Recuva* o *Parted Magic*.

R Partiamo dal presupposto che **Recuva** è un programma per Windows, mentre **Parted Magic** è un software commerciale che in questo tipo di operazioni non ha grandi motivi per essere utilizzato. In Linux, per fortuna, ci sono già una serie di strumenti che possono fare al caso nostro. L'unico requisito richiesto è l'uso della riga di comando. Non temete però, perché questa modalità non ha niente di spaventoso. Richiede solo un approccio diverso rispetto alla tradizionale interfaccia grafica. La prima cosa da fare quando si eliminano erroneamente dei file, è smettere di scrivere sul disco fisso. Se infatti non si sovrascrive alcunché sui file cancellati, tutto quello che avrete fatto sarà eliminare i loro riferimenti dall'indice del filesystem. In altre parole, togliete solo un'etichetta, mantenendo intatti i dati. Attenzione però, perché lo spazio occupato da questi file viene contrassegnato come disponibile alla scrittura.
Se perseverate nell'utilizzo del sistema, finirete per eliminarli definitivamente.
Se avete spazio a sufficienza su un'altra unità o una partizione (non un'altra cartella nello stesso filesystem), allora potete fare un backup di tutto con il comando **dd**:

```
dd bs=4k if=/dev/sda2 of=/media/driveesterno/backup.img
```

Sarà quindi possibile utilizzare l'immagine per recuperare ciò che avete perso. Se utilizzate un filesystem extundelete ext3 o ext4, disponibile nei repo della maggior parte delle distribuzioni, potete sfruttare un filesystem smontato o un file di immagine. L'opzione per ripristinare il tutto è:

```
extundelete /media/driveesterno/backup.img --restore-all
```

oppure

```
extundelete /dev/sda2 --restore-all
```

Tutti i file cancellati trovati, a questo punto, verranno ripristinati in una directory chiamata **RECOVERED_FILES**. Se volete collocarli in una cartella specifica, potete utilizzare:

```
extundelete /media/driveesterno/backup.img --restore-directory percorso/per/directory
```

Il percorso è relativo alla radice del filesystem. Se quindi si dispone di una partizione **home** separata, potreste dover usare **username/datiimportanti**.
Se **home** è nella partizione di root, niente vi impedisce di utilizzare **home/username/datiimportanti**. Per scoprire le altre opzioni di extundelete, usate il comando:

```
extundelete --help
```

Se extundelete non risolve il problema, potete provare **PhotoRec**. Il ruolo di questo software è analizzare l'intero filesystem, salvando tutti i file che trova. A causa del suo funzionamento e della rimozione delle voci dalla tabella del filesystem, PhotoRec recupera solo il contenuto dei file, ma non i metadati. Questi includono le proprietà dei singoli documenti, ma soprattutto il nome e le autorizzazioni. Per capire cosa dovrete recuperare, quindi, servirà un po' di tempo. Se per esempio i file andati persi sono di un tipo specifico, come JPEG, provate con:

```
file * | grep JPEG
```

4 Problemi di root

Sono recentemente approdato al mondo Linux dopo una lunga militanza con Windows. Sto imparando i rudimenti dei comandi e nel tentativo di installare un antivirus con *apt-get*, mi vengono chiesti i privilegi di root. Da quello che ho capito, si tratta di una specie di modalità superutente. In realtà io pensavo di avere già eseguito il log-in come amministratore del sistema.

Il root può essere paragonato alla modalità amministratore di Windows, tranne per il fatto che si tratta di un utente separato. Linux, infatti, è un sistema operativo multi-utente, dove ogni user ha un proprio spazio di archiviazione a disposizione. L'utente root, o superuser, è l'unico a essere autorizzato all'installazione di software nel sistema. In altre parole, per aggiungere qualsiasi programma alla distro Linux con **apt-get**, dovete essere collegati come root. A differenza dei sistemi operativi Windows, con Linux è possibile elevarsi solo temporaneamente ad amministratore di sistema.
Il comando per farlo è **sudo** che consente di installare velocemente qualsiasi applicazione, per poi tornare ai diritti di utente normale una volta finita la procedura. In questo modo, la sicurezza del sistema viene mantenuta ad alti livelli, limitando la possibilità di fare qualche danno nel caso doveste scordarvi di eseguire il logout dal superutente.
Per eseguire correttamente il comando **apt-get**, quindi, bisogna anteporre l'istruzione sudo. Inserendo poi la password si potrà procedere a installare il software in questione. Per quanto riguarda la necessità di aggiungere un antivirus a Linux, tenete presente un fattore importante. Le minacce che colpiscono questo sistema sono davvero poche. Certo, installare un software di questo tipo non comporta alcun problema e in definitiva può fornire un baluardo contro eventuali attacchi, ma non dovete commettere l'errore di scambiare Linux per Windows.

5 Patch al kernel

Vorrei sapere come si applica una patch al kernel e soprattutto se si tratta di un'operazione consigliabile.

Applicare una patch al Kernel permette di correggere eventuali problemi senza scaricare l'intero pacchetto. La procedura è quindi consigliabile nei casi in cui, oltre alla disponibilità della specifica patch, ci sia la necessità di risparmiare tempo. Infatti, evitate di effettuare il download di svariate centinaia di Megabyte, fermandovi al solo componente che volete. Per la sua applicazione, la prima cosa da fare è copiare il file in questione all'interno della directory **/usr/src/**, quindi accedere al percorso da terminale. Qui, inserite il comando:

```
bzip2 -dc nomefile.bz2 |patch -p1
```

oppure:

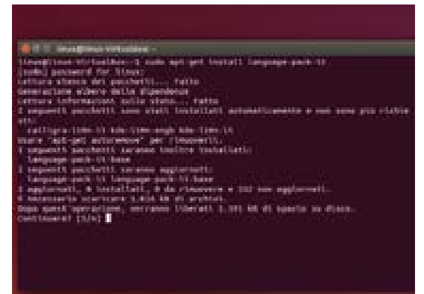
```
gzip -dc nomefile.gz |patch -p0.
```

nel caso la patch sia in formato gzip.

Soluzione rapida

Quando installiamo Ubuntu o le sue derivate, può succedere che il supporto per le lingue non venga montato correttamente. Non è inusuale che nonostante la nostra scelta principale ricada sull'italiano, spesso il sistema compaia con istruzioni ancora in inglese. Per risolvere il problema, in questo caso, la soluzione migliore è provvedere alla reinstallazione del language pack che vi interessa. Per farlo, aprite semplicemente il terminale e scrivete:

```
sudo apt-get install language-pack-it
```



In alternativa, è possibile utilizzare anche l'eseguibile **patch-kernel** presente nella directory **/usr/src/linux/script/**.
Si tratta di uno script utile a recuperare i file della patch e capace di eseguirli automaticamente. Dopo aver scompattato il componente nel percorso **/usr/src/**, usate il seguente comando per avviare il patch-kernel:

```
cd /usr/src/linux/scripts/patch-kernel
```

Patch completata con successo! LXP



FAQ: le domande più frequenti

Usare *su*

› Alcune distribuzioni usano sudo, mentre altre utilizzano il più tradizionale *su*. Quale dei due è il migliore?

In questo caso non si può parlare di migliore o peggiore, quanto più della diversità nell'approccio. **sudo** è perfetto per dare accesso root a un utente che vuole svolgere un compito specifico, senza peraltro fornirgli la password di amministrazione. Per esempio quando vuole installare un'applicazione. **su**, invece, permette l'accesso di un utente ai permessi di un altro (di solito root).

› Bene, ho capito. Se però io voglio pieno accesso come superutente, su è la scelta più ovvia?

Non necessariamente. su, infatti, non sta per *Super Utente* (*Super User*), ma per *Switch User*. Può essere utilizzato per fare in modo che un utente abbia gli stessi permessi di un altro. Il valore predefinito con cui viene utilizzato è root, ma non è l'unico. Ponendo che come amministratore di sistema tu voglia accedere all'utente Mario, magari per correggere qualche problema, puoi farlo con su. In questo caso, passando da root a semplice utente, non hai bisogno di alcuna password.

› Molto interessante, ma non ho ancora capito come avere accesso completo al root. Torno quindi a chiedere se su è il comando giusto per fare quanto chiedo.

Sì e no. Nel senso che su ti dà solo i privilegi di superuser, ma durante il loro utilizzo ti troverai ancora all'interno delle specifiche utilizzate dall'utente principale. Non sarai quindi in un ambiente root. Tutte le variabili impostate, insieme alla directory corrente, rimangono le stesse e il percorso viene resettato a **/sbin:/bin:/usr/sbin:/usr/bin**. Tutto quanto è stato installato in **/user/local**, quindi, non è più direttamente disponibile.

› Adesso è tutto chiaro. Quindi, come faccio a ottenere un pieno accesso root da un terminale X sul desktop dell'utente?

Aggiungi **-** al comando su. Questo, però, deve comparire in tutte le altre opzioni. È inoltre possibile utilizzare **--login**, peraltro più leggibile, ma più fastidioso da scrivere. L'uso di questi due comandi dipende essenzialmente dal tuo modo di pensare: sei un programmatore Perl o un compilatore Phyton? L'opzione **-** offre un ambiente libero dal log-in al 99% e corrisponde essenzialmente a un accesso diretto come root o a qualsiasi altro utente specifico.



L'eco dei LUG

I Lug

I LUG rappresentano da sempre il punto di riferimento per chiunque voglia conoscere GNU/Linux. Ogni mese dedicheremo loro questo spazio per la comunicazione di nuovi progetti e appuntamenti. Se hai qualcosa da segnalarci scrivi a **ecodeilug@linuxpro.it**

ABRUZZO

AnxaLUG - Lanciano
www.anxalug.org
Il Pinguino - Teramo
Non disponibile
MarsicaLUG - Marsica
www.marsicalug.it
OpenLUG - L'Aquila
Non disponibile
Pescara LUG
www.pescaralug.org
Pineto LUG
www.pinetolug.org
Pollinux LUG - Pollutri
Non disponibile
SSVLUG - San Salvo, Vasto, Termoli
www.ssvlug.org
SulmonaLUG
http://sulmonalug.it
TeateLUG - Chieti
Non disponibile
TeLUG - Teramo
www.telug.it
User Group Valle Roveto
http://linuxvalley-os4.blogspot.com/

BASILICATA

Basilicata LUG - Potenza e Matera
www.baslug.org

CALABRIA

3BYLug - Trebisacce
www.3bylug.tk
Bogomips - Bisignano
www.blug.it
CastroLUG
http://castrolug.altervista.org
Cosenza Hack Laboratory
http://hacklab.cosenzainrete.it/
CSLUG - Cosenza
http://cslug.linux.it
CzLug
Non disponibile
HackLab Catanzaro
http://hacklab.cz
Piana LUG - Piana di Gioia Tauro
Non disponibile
Reggio Calabria LUG
http://rclug.linux.it
Revolutionary Mind
www.revolutionarymind.org
SpixLug - Spezzano Albanese
Non disponibile

CAMPANIA

AFR@Linux LUG
www.afralinux.netsons.org
Afralug - Afragola
www.afralug.com
CasertaLUG
www.casertaglug.org
Hackaserta 81100
www.81100.eu.org
HackMeetNaples Napoli HackLab
www1.autistici.org/hmn
IGLUG - Napoli e provincia
www.iglug.org
IRLUG - Irpinia
www.irlug.it
LUG-Ischia
www.lug-ischia.org
NALUG - Napoli
www.nalug.net
Neapolis Hacklab
www.officina99.org/hacklab.html
Padulug - Paduli (BN)
http://linux.paduli.com
SCALUG - Scafati (SA)
http://xoomer.alice.it/scalug/
Tuxway.org - Provincia di Napoli
www.tuxway.org
VaLug - Vallo Linux User Group
www.valug.it
XALUG - Salerno
http://xalug.tuxlab.org

EMILIA ROMAGNA

ALFLUG - Alfonsine
www.alflug.it
Borgotaro LUG - Val Taro
http://btlug.it/
ConoscereLinux - Modena
www.conoscerelinux.it
ERLUG
http://erlug.linux.it
Ferrara LUG
www.ferrara.linux.it
FoLUG - Forlì
http://folug.linux.it
ImoLUG - Imola
www.imolug.org
LUGPiacenza
www.lugpiacenza.org
PANLUG - Vignola
Non disponibile
PLUG - Parma
http://parma.linux.it
RavennaLUG
www.ravennalug.org
RELug - Reggio Emilia e provincia
http://relug.linux.it
RiminiLug
www.riminilug.it
S.P.R.I.Te
http://sprite.csr.unibo.it
UIELinux - Valle del Rubicone
www.uielinux.org

FRIULI VENEZIA GIULIA

GOLUG - Gorizia
www.golug.it
IGLU - Udine
http://iglu.cc.uniud.it
LUG Pordenone
www.pnlug.it
LugTrieste
http://trieste.linux.it
LUG [A] [L] [P] - Aquileia
www.alproject.org

LAZIO

CiLUG - Frosinone
www.cilug.org
CLUG - Cassino
http://cassino.linux.it/
GioveLUG - Terracina
www.giovelug.org
La Sapienza LUG
www.lslug.org
Latina LUG
www.llg.it
LUG Privernum Volsca - Priverno (LT)
www.pvlug.org
LUGRoma
www.lugroma.org
LUGRoma 3
www.lugroma3.org
TorLUG - Università Tor Vergata - Roma
http://lug.uniroma2.it/
V.I.S.C.O.S.A. - Ciampino
www.viscosa.org

LIGURIA

Genuense Lug - Genova e d'intorni
http://genova.linux.it
GinLug - Genova Sampierdarena
www.sennaweb.org
Govonis GNU/LUG - Provincia di Savona
www.govonis.org
SLIMP - Software Libero Imperia
http://slimp.it/
TLug-TSL - Tigullio Ligure
http://tlug.linux.it/

LOMBARDIA

BGLug - Bergamo e provincia
www.bglug.it
BGLug Valle Seriana - Valle Seriana
http://bglugvs.web3king.com/
GL-Como - Como
www.gl-como.it
GLUX - Lecco e provincia
www.lecco.linux.it
GULLP - Gruppo Utenti Linux Lonate Pozzolo
www.gullp.it
IspraLUG - Ispra
http://ispralug.eu/
LIFO - Varese
www.lifolab.org
LIFOS - Cinisello Balsamo
www.lifos.org
Linux Var - Varese
www.linuxvar.it
LoLug - Lodi e provincia
www.lolug.org
Lug Bocconi - Milano
www.lug-bocconi.org
LugBS - Brescia e provincia
http://lugbs.linux.it/
Lug Castegnato - Castegnato
www.kenparker.eu/LugCastegnato
LugCR - Cremona e provincia
www.lugcr.it
Lug Crema - Crema
http://filibusta.crema.unimi.it/
LUGDucale - Vigevano

L'eco dei LUG

www.lugducale.it
LugMan - Mantova e provincia
www.lugman.org
LugOB - Cologne e ovest bresciano
www.lugob.org
MoBLUG - Monza e Brianza
www.bubblesfactory.it
OpenLabs - Milano
www.openlabs.it
POuL - Milano
www.poul.org
TiLug - Pavia
http://pavia.linux.it
VigLug - Vignate, Milano Est - Adda Martesana
www.viglug.org

MARCHE
Ascolinux LUG/FSUG Ascoli
http://marche.linux.it/ascoli/
CameLUG - Camerino
www.camelug.it
CMlug
www.cmlug.org
Egloo
www.egloo.org
FanoLUG
www.fanolug.org
Fermo LUG
www.linuxfm.org/fermolug/
GLM - Macerata
www.gruppolinuxmc.it/start/index.php
LUG Ancona
www.egloo.org
LUG Jesi
www.lugjesi.net
LUG Marche
http://marche.linux.it
PDP Free Software User Group
http://pdp.linux.it
SenaLug - Senigallia
www.lug.senigallia.biz

MOLISE
Campobasso LUG
http://cb.linux.it/
FrenterLUG - Larino
Non disponibile
SmaLUG - San Martino
www.smalug.org

PIEMONTE
ABC Lug - Alba/Bra/Carmagnola
http://abc.linux.it/
AlLug - Alessandria e provincia
www.allug.it
BiLUG - Provincia di Biella
http://www.bilug.it
FASoLi - Alessandria e provincia
http://softwarelibero.al.it/
Gallug - Galliate
www.gallug.it
GlugTO - Torino e provincia
www.torino.linux.it
IvLug - Ivrea Linux User Group
www.ivlug.it
Linox Novara
www.linoxnovara.org
SLIP - Pinerolo
http://pinerolo.linux.it/
ValSusinux - Val Susa e Val Sangone
www.valsusinux.it

PUGLIA
BriLUG - Brindisi
www.brilug.it
CapitanLUG - Capitanata
www.capitanlug.it
LATLUG - Latiano Linux User Group
www.latlug.org
LUGargano
www.lugargano.it
LUGBari - Bari e provincia
www.lugbari.org
MurgiaLug - Santeramo in Colle
www.open-pc.eu/index.php/murgialug/
SaLUG! - Salento
http://salug.it
Talug - Taranto
www.talug.it

SARDEGNA
GNUraghe - Oristano
www.gnuraghe.org
GULCh - Cagliari
www.gulch.it
PLUGS - Sassari
www.plugs.it

SICILIA
CefaLug - Cefalù
http://cefalug.linux.it
cLUG - Caltanissetta
www.clug.it
EnnaLUG
www.ennalug.org
FreakNet MediaLab - Catania
www.freaknet.org
Leonforte LUG
http://leonforte.linux.it
LUG Catania
www.catania.linux.it
LUGSR - Siracusa
www.siracusa.linux.it
MELUG - Messina
Non disponibile
Norp LUG - Noto, Pachino, Rosolini
Non disponibile
PALUG - Palermo
http://palermo.linux.it
RgLUG - Ragusa e provincia
http://ragusa.linux.it
VPLUG Linux Planet - Provincia Caltanisetta
www.vplug.it
SputniX - Palermo
www.sputnix.it

TOSCANA
ACROS - Versilia, Lucca, Massa Carrara
www.lug-acros.org
Elbalinux
Non disponibile
ElsaGLUG - Val d'Elsa
www.elsaglug.org
FLUG - Firenze
www.firenze.linux.it
GOLEM - Empoli, Valdelsa
http://golem.linux.it
GroLUG - Grosseto
www.grolug.org
G.U.L.LI - Livorno
www.livorno.linux.it
GulP! Piombino
http://gulp.perlmonk.org
GULP Pisa
www.gulp.linux.it
GuruAtWork - Grosseto e provincia
www.guruatwork.com
IPIOS - Bibbiena e valle del Casentino
www.ipios.org
Lucca LUG
http://luccalug.it
L.U.G.A.R - Arezzo
Non disponibile
PLUG - Prato e provincia
www.prato.linux.it
PtLug - Pistoia e provincia
www.ptlug.org
SLUG - Siena e provincia
www.siena.linux.it

TRENTINO ALTO ADIGE
LinuxTrent - Trento
http://linuxtrent.it
LugBz - Bolzano
www.lugbz.org

UMBRIA
OrvietoLUG
www.orvietolug.it
LUG Perugia
www.perugiagnulug.org
TerniLUG
www.ternignulug.org

VALLE D'AOSTA
SLAG - Aosta
www.slag.it

VENETO
0421ug - Provincia di Venezia
www.0421ug.org
BLUG - Belluno
http://belluno.linux.it
Faber Libertatis - Padova
http://faberlibertatis.org
GrappaLUG - Bassano del Grappa
http://grappalug.homelinux.net/
ILC - Informatica Libera Cittadellese - FSUG
http://ilc.pd.it
LegnagoLUG
Non disponibile
Linux Ludus - Villafranca (VR)
www.linuxludus.it
LugAnegA
www.luganega.org
LUGSF - San Fidenzio
Non disponibile
LUG Vicenza
www.vicenza.linux.it
LugVR - Verona
www.verona.linux.it
MontelLUG - Montebelluna
www.montellug.it
FSUG Padova
www.fsugpadova.org
RoLUG - Rovigo
http://rovigo.linux.it
TVLUG - Treviso
www.tvlug.it
VELug - Venezia
www.velug.it
AViLUG Schio
http://www.avilug.it/doku.php

NAZIONALI
FSUGitalia
www.fsugitalia.org
Gentoo Channel Italia
www.gechi.it
MajaGLUG
www.majaglug.net
SkyLUG
http://tech.groups.yahoo.com/group/skylug/

Guida DVD

Ogni mese Linux Pro vi offre i programmi e le distribuzioni più recenti su DVD



Ogni volta che troverete questo simbolo in un articolo, vorrà dire che i file citati si trovano nel DVD allegato alla rivista.

Distro Desktop

Innovazione al potere: Fedora 21

Sul DVD di questo numero trovate una distro nota per le sue innovazioni. In queste pagine parleremo in modo generico di Fedora ma per gli aspetti più peculiari di questa distro vi consigliamo di leggere l'articolo che le abbiamo dedicato a pagina 22. Nel DVD allegato alla rivista troverete le tre versioni di **Fedora 21**: **workstation**, **server** e **cloud**. Destinate a usi specifici, queste release segnano un importante passo in avanti nel processo di creazione di una distro. Non una, ma tre piattaforme basate sullo stesso core, ma con strumenti e personalizzazioni diverse.

Workstation e Server

Fedora 21 Workstation è pensato per essere utilizzato con i computer desktop e offre un insieme completo di programmi per ogni esigenza. La versione Server, invece, è pensata per l'uso nei datacenter o in tutte quelle situazioni in cui avete la necessità di controllare fin nei minimi particolari un'infrastruttura di rete. La release Cloud, invece, è stata sottoposta a un'intensa cura dimagrante, così da ridurre al minimo il peso complessivo del sistema. Può essere utilizzata in ambienti cloud dove ciò che conta è avere solo gli strumenti indispensabili per eseguire un certo compito. Tutte e tre le versioni sono disponibili sia per macchine a 32 bit sia a 64 (noi, per motivi di spazio, abbiamo incluso nel DVD solo le ISO per piattaforme x86). Fedora 21 Workstation si distingue fin da subito per l'elegante interfaccia utente in stile Gnome 3. Offre tutto quello che serve sia per la produttività in uso quotidiano sia per gli sviluppatori. Nel primo caso, abbiamo la suite completa di LibreOffice, ma anche programmi come Empathy, Gedit, Shotwell, Rhythmbox e altri. Per i programmatori che vogliono provare il proprio codice su diverse piattaforme, ma anche per chi desidera semplicemente mettere mano ad altri sistemi senza installarli su PC, c'è **Boxes**. Si tratta di un programma progettato per la creazione di macchine virtuali. Semplice e intuitivo da utilizzare, basta seguire le istruzioni passo a passo per rendere operativa una virtual machine con qualsiasi sistema. Fedora 21 Server segue la stessa filosofia di semplicità e immediatezza della controparte per workstation. Tra gli strumenti più importanti e utili troviamo **Cockpit**. Questa interfaccia Web consente di controllare in modo veloce e facile l'intera infrastruttura di rete. Possiamo gestire le prestazioni e lo stato dei sistemi, eseguire il deploy e supervisionare i servizi basati su container. Il tutto facendo uso di una GUI che rende le attività molto più flessibili. Se poi abbiamo la necessità di configurare diversi server, Fedora 21 ci viene in aiuto con i ruoli **server** e **rolekit**. Con quest'ultimo creiamo, gestiamo e distribuiamo i ruoli che si basano sulla piattaforma Fedora server e consentono di sfruttare una base per la configurazione di una macchina di rete. Un altro importante strumento messo a disposizione da questa distro è il gestore d'identità FreeIPSA. Esso consente di configurare utenti, sistemi e policy tramite un ambiente strutturato per essere il più immediato e intuitivo possibile.

Fedora Cloud

La terza versione di Fedora, vale a dire Cloud, si distingue per le ridotte dimensioni che consentono di utilizzarla in qualsiasi ambiente basato sulla nuvola. Ne esistono due release: per uso generico e per Container. Nel primo caso, abbiamo un'immagine di solo 100 MB, nel secondo di 232. La distro per uso generico è stata pensata per la creazione di macchine virtuali, mentre quella per Container offre tutti gli strumenti per l'uso LXC. Infine, per tutti coloro che vogliono dare un'occhiata a come sarà Fedora 22, nel momento in cui scriviamo, potete scaricare la Alpha e la Beta release all'indirizzo **http://bit.ly/1lmldut**.

Cosa c'è nel DVD

- Audex 0.79
- Bash 4.3.30
- Chocolate Doom 2.1.0
- ClamAV 0.98.6
- Debreate 0.7.10
- Fifth 0.2
- Fedora 21
- GeexBox 3.1
- GParted Live 0.22
- Hello 2.10
- Lightweight Sec. 1.5.6
- LinHES 8.3
- Linux Console 2.3
- RecomXaraLX 0.7
- newLISP 10.6.2
- Perl 5.20.2
- Python 3.5.0a3
- QuiteRSS 0.17.6
- Red Eclipse 1.5
- SystemRescueCD 4.5
- Speed Dreems 2.1
- Untangle NG Firewall
- xDecoration 0.1.7
- Xfdesktop 4.10.3

Cosa ci offre Fedora 21 Workstation



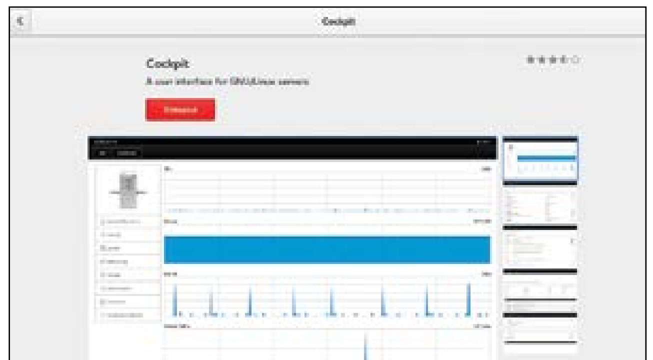
1 Desktop ordinato
L'ordine e la praticità del desktop Gnome sono inconfondibili. Avete tutti gli strumenti subito a disposizione e nel caso voleste restringere il campo, basta utilizzare il motore di ricerca interno. Veloce e preciso, consente di trovare tutto quello di cui avete bisogno.



2 Virtual Machine al volo
La creazione di una macchina virtuale con **Boxes** è un gioco da ragazzi. Tutto ciò che dovete fare è seguire la procedura passo a passo che vi guida nella creazione di una virtual machine. Il primo passaggio consiste nell'inserire il supporto d'installazione del sistema operativo.



3 Software center
Gnome Software mette a disposizione un ambiente pratico e semplice da cui scaricare e installare tutte le applicazioni di cui avete bisogno. Per trovare i vari software, potete usare il motore di ricerca integrato, oppure navigare tra le varie categorie presenti.



4 Cockpit anche per workstation
Cockpit, lo strumento per la gestione server di più piattaforme, non è integrato in Fedora Workstation. Tuttavia niente impedisce di scaricarlo da Gnome Software e utilizzarlo anche all'interno di questa distro, sebbene il suo ambiente naturale sia la versione Server di Fedora.



5 Che tempo fa?
Tra le varie applicazioni di Fedora 21 troviamo anche **Meteo**. Quest'ultima offre una panoramica precisa delle condizioni meteorologiche di un dato luogo. Non dovete far altro che inserire la località in fase di accesso. Se poi volete modificarla, basta che premiate il pulsante **Luoghi**.



6 Supporto per sviluppatori
Il **Developer Assistant**, abbreviato in **DevAssistant**, permette agli sviluppatori di gestire gli aspetti secondari della programmazione. Per creare un progetto in Python GTK+3, per esempio, basta premere il pulsante **Python** e selezionare la relativa opzione.

In edicola il 12 giugno

NEL PROSSIMO NUMERO

Hai un argomento da proporci?
Scrivi a prossimamente@linux.it

100 STRUMENTI INDISPENSABILI

Abbiamo raccolto e messo alla prova le migliori applicazioni per Linux e abbiamo compilato la lista di quelle che devi assolutamente avere!



Mensile, prezzo di copertina 5,90 €
www.linuxpro.it

Direttore responsabile: *Luca Sprea*

Traduzione e Localizzazionea cura di:
Ventidodici di Andrea Orchesi
redazione@linuxpro.it


Sprea S.p.A.
Socio Unico - direzione e coordinamento di Sprea Holding S.p.A.

Presidente: Luca Sprea
Consigliere delegato: Mario Sprea

Coordinamento:
Gabriella Re (Foreign Rights) international@sprea.it, Alberta Rivolta (PA), Ambra Palermi (Segreteria Editoriale), Francesca Sigismondi (Ufficio Legale), Tiziana Rosato (acquisti e produzione), Emanuela Mapelli (Pianificazione Pubblicitaria)

Amministrazione: Anna Nese (CFO), Erika Colombo (controller), Irene Citino, Sara Palestra
amministrazione@sprea.it

Servizio qualità edicolanti e DL: Sonia Lancellotti, Andrea Palermi
distribuzione@sprea.it

PUBBLICITA' E MARKETING:
Alessandra Cappellacci- alessandracappellacci@sprea.it
Tel.0292432275 - Cell.3204670523

Sede Legale: - via Torino, 51 20063 Cernusco Sul Naviglio (Mi) - Italia
PI 12770820152- Iscrizione camera Commercio 00746350149

ABBONAMENTI E ARRETRATI

Servizio qualità abbonamenti e arretrati: Desirée Conti

Abbonamenti: si sottoscrivono on-line su www.linuxpro.it/abbonamenti
Mail: abbonamenti@linuxpro.it
Fax: 02 700 53 76 72
Tel: 02 87 15 82 25 (lun-ven / 9:00-18:00)

Arretrati: si acquistano on-line su www.linuxpro.it/arretrati
Mail: arretrati@linuxpro.it
Fax: 02 700 53 76 72 - Tel: 02 87 15 82 25 (lun-ven / 9:00-18:00)
Il prezzo dell'abbonamento è calcolato in modo etico perché sia un servizio utile e non in concorrenza con la distribuzione in edicola.

www.myabb.it a cura di Aktia srl
Contenuti su licenza: "Linux Format"- Future Publishing Limited PLC., Bath UK

Registrazione testata: Linux Pro, pubblicazione mensile registrata al Tribunale di Milano il 08.02.2003 con il numero 74.

Distributore per l'Italia e per l'estero:
Press-Di Distribuzione stampa e multimedia s.r.l. - 20134 Milano
ISSN: 1722-6163

Stampa: Arti Grafiche Boccia S.p.A.- Salerno